Université de Provence,

Département Sciences, Arts et Techniques
de l'Image et du Son SATIS

L'enjeu des logiciels libres : Création d'une Formation M.A.O. sous Linux.

Mémoire de Master 2 Professionnel

Présenté par Erwan INYZANT

Sous la direction de Monsieur Rémi Adjiman

Année universitaire 2007-2008

Université de Provence,

Département Sciences, Arts et Techniques
de l'Image et du Son SATIS

L'enjeu des logiciels libres : Création d'une Formation M.A.O. sous Linux.

Mémoire de Master 2 Professionnel

Présenté par Erwan INYZANT

Sous la direction de Monsieur Rémi Adjiman

Année universitaire 2007-2008

Université de Provence,

Département Sciences, Arts et Techniques
de l'Image et du Son SATIS

L'enjeu des logiciels libres : Création d'une Formation M.A.O. sous Linux.

Mémoire de Master 2 Professionnel

Présenté par Erwan INYZANT

Sous la direction de Monsieur Rémi Adjiman

Année universitaire 2007-2008

Remerciements :

J'aimerais remercier toutes ces personnes qui m'ont soutenu et aidé pendant la rédaction de ce mémoire et sans qui il m'aurait été difficile de le finir :

Rémi Adjiman
pour ses conseils son soutien et sa disponibilité
tout au long de ce projet

Didier Merlateau et Rolf Entgelmeier
pour m'avoir accordé de leur temps pour une interview
et partagé leur expérience de formateur

Yohann Béréziat et Florent Berthaut
pour avoir accepté de témoigner en tant
qu'utilisateur de Linux pour la création musicale.

Toute l'équipe de LinuxMAO
pour leur disponibilité, et le travail formidable qu'ils font

Mes parents
pour leur soutien moral, la relecture et
la correction des fautes

Thomas Thiery
pour ses conseils judicieux.

Ma femme
pour m'avoir supporté pendant toute
la durée de ce projet

Et toute l'équipe des enseignants de SATIS.
pour ce qu'ils m'ont apporté,
et sans qui cela ne serait jamais arrivé.

L'enjeu des logiciels libres : Création d'une Formation M.A.O. sous Linux.

Résumé :

Les logiciels libres font petit à petit partie du paysage informatique. Déjà bien implantés dans certains domaines tels que les serveurs et l'administration de systèmes informatisés, ils commencent doucement à se démocratiser auprès du grand public, notamment grâce à Linux. En effet ce système d'exploitation devient une solution de plus en plus abordable pour un utilisateur lambda. OpenOffice pour la Bureautique, Firefox et Thunderbird pour Internet et la messagerie, rien ne fait défaut. Ces logiciels sont déjà utilisés par certaines administrations françaises. Le multimédia et l'Audio ne sont pas épargnés et en l'espace de quelques années, de nombreux logiciels ont vu le jour et ont porté le système d'exploitation Linux à un niveau tel, qu'il est envisageable de l'utiliser pour la création musicale. Les évènements comme Linux Audio Conference (LAC) qui a lieu une fois par an et qui rassemble tous les acteurs (chercheurs, développeurs, musiciens, utilisateurs) de l'Audio sous Linux, témoignent de cet engouement.

Ce mémoire s'articule autour de 2 grands thèmes. Le premier a pour but d'explorer les logiciels libres : historique, philosophie, différences avec les logiciels propriétaires, les licences libres. Le deuxième a un objectif plus professionnel. En effet, c'est une réflexion sur la manière de créer et d'organiser une formation en Musique Assistée par Ordinateur utilisant exclusivement des logiciels libres. Après une présentation de Linux et de ses différents composants indispensables pour la création musicale, je détaille cette réflexion tant sur le plan de son contenu que de son organisation et de la pédagogie à adopter. Le premier module est disponible sous licence Creative Commons. Il peut être consulté et téléchargé sur le site http://mysthr.free.fr/Joomla (section Guide Audio).

Mots clés:

Logiciels libres ; Open Source ; Linux ; M.A.O. ; formation ; licences libres ; Fedora ; PlanetCCRMA

Table des matières

Introduction....11

I. Les Logiciels Libres....13
I.1 - La naissance d'une culture du libre....13
a) L'anecdote historique....13
b) Le projet de Stallman....13
c) Rencontre avec Linus Torvalds....14
I.2 - Qu'est ce qui fait le succès du libre ?....16
a) Exemples de logiciels....17
b) Et plus largement :....17
c) Les distributions Linux....19

II. Les licences : Explication / Exemples....22
II.1 - Pour le logiciel....22
a) État des lieux....22
b) La GNU GPL : GNU General Public Licence....23
c) La GNU LGPL (Lesser General Public Licence) 24
II.2 - Au delà du logiciel :....24
a) GNU Free Documentation Licence :....25
b) Licence Art Libre (LAL) :....25
c) Creative Common :....26
d) Petit bémol....28

III. Le Libre et l'Audio....29
III.1 - Pourquoi utiliser des logiciels libres ?....29
a) Éthique....29
b) Libre accès....29
c) Indépendance et pérennité....29
d) Qualités techniques....30
e) Communauté du Logiciel Libre....30
f) Éviter l'utilisation de formats fermés....30
g) Les limites des logiciels libres ?....31
III.2 - Un système d'exploitation libre....32
a) Les bases de l'audio sous Linux....33
b) Les logiciels....36

IV. Création d'une Formation M.A.O....40
IV.1 - But de la formation : Pourquoi, pour qui, comment ? 40
a) Pourquoi ?....40
b) Pour qui ?....41
c) Le programme....42
d) La pédagogie....44
e) Mise en place d'une formation....45
IV.2 - Module 1 : Installation et configuration du système....46
a) Choix de la distribution....46

b) Installation du système d'exploitation Linux. Dual boot avec Windows ou Mac.....47
c) Maintenance du système : Organisation de l'arborescence et des fichiers, le terminal, configuration système.....47
d) Paramétrage et installation des logiciels/paquets nécessaires, et du noyau temps réel.....48
e) Les drivers ALSA/Freebob : présentation, configuration.....49
f) Derniers petits détails.....49
g) Présentation et configuration du serveur audio JACK.....49
IV.3 - Rencontres avec des professionnels.....50
a) Explication de la démarche.....50
b) Didier Merlateau.....50
c) Rolf Entgelmeier de l'AGESCA.....51
d) Bilan des rencontres.....52

Conclusion.....54

Annexes.....55
Annexe 1 : Le Manifeste GNU.....55
Annexe 2 : Témoignages d'utilisation de Linux en M.A.O.....65
Introduction.....65
a) Xrousse (Yohann Béréziat).....65
b) Hitmuri (Florent Berthaut).....67

Netographie.....68

Introduction

Ce mémoire se veut être une exploration de ce que sont les logiciels libres en vue de la création d'une formation en Musique Assistée par Ordinateur sous Linux. A l'heure où l'informatique est omniprésente dans notre vie, on entend parler de plus en plus de logiciels libres. Cette notion qui n'existait quasiment pas il y a une dizaine d'année, se répand dans le monde informatique, notamment grâce au boum de l'Internet à haut débit. Ils s'appuient sur un concept existant depuis des centaines d'années. Voici une citation de Victor Hugo qui pourrait s'appliquer à ces logiciels et qui résume bien la philosophie de ces développeurs :

> *« Le livre, comme livre, appartient à l'auteur, mais comme pensée, il appartient - le mot n'est pas trop vaste - au genre humain. Toutes les intelligences y ont droit. Si l'un des deux droits, le droit de l'écrivain et le droit de l'esprit humain, devait être sacrifié, ce serait, certes, le droit de l'écrivain, car l'intérêt public est notre préoccupation unique, et tous, je le déclare, doivent passer avant nous »*
> *Victor Hugo*[a]

Qu'est ce qu'un logiciel libre ? Qu'apporte-il ? Quelle crédibilité peut on lui accorder ? Autant de questions auxquelles je tenterai de répondre dans une première partie destinée à rappeler l'origine de ces logiciels et à introduire cette notion de liberté. J'étendrai mon propos sur les licences libres à d'autres projets se basant sur le même modèle, tels que les encyclopédies collaboratives, la musique et l'art en général. En effet ces nouvelles licences ont ouvert de nouvelles possibilités de partage, de diffusion et de collaboration.

Après cette présentation sur les enjeux que peuvent représenter les logiciels libres, je présenterai le système d'exploitation Linux qui me servira de base à la formation en Musique Assistée par Ordinateur (M.A.O.). En effet, une des perspectives de ce mémoire est de créer une formation M.A.O. en utilisant exclusivement des logiciels libres. La plateforme Linux est donc la plus appropriée. Je détaillerai ce système d'exploitation relativement peu connu, mais qui offre de nombreuses possibilités : explication de son fonctionnement, présentation des logiciels audio.

La mise en place d'une formation n'est pas simple et soulève de nombreuses questions. Il faut bien définir ses objectifs, le public visé, proposer un contenu pertinent et organisé de manière logique. La pédagogie est aussi un aspect important, car elle est un point clé dans la transmission des connaissances et un enseignement efficace. Le but de ma partie pratique est de proposer une formation simple et accessible au plus grand nombre, répondant à toutes ces problématiques afin d'établir un programme détaillé.

Enfin, ce mémoire est aussi une manière de faire le point sur les possibilités qu'offrent les logiciels libres en matière de création musicale. Une des questions sera donc de savoir s'il est judicieux de proposer une telle formation sur un marché déjà bien encombré.

---

a Cité par Jean-Pierre Archambault pendant une conférence sur les logiciels libres et l'éducation. Cette citation est un extrait du « Discours d'ouverture du Congrès littéraire international de 1878 » de Victor Hugo.

I. Les Logiciels Libres

I.1 - La naissance d'une culture du libre.

a) L'anecdote historique[a]

L'anecdote historique raconte qu'au laboratoire d'intelligence artificielle de l'Institut de Technologie du Massachusetts (M.I.T.), une imprimante tombait souvent en panne. Mais comme les chercheurs disposaient du code source[1] du pilote[2], ils avaient modifié le programme pour que l'imprimante leur envoie un signal à chaque panne. Entre-temps, l'équipe du laboratoire avait développé un système d'exploitation informatique à temps partagé. Ils autorisaient l'utilisation de ces logiciels par d'autres universitaires ou ingénieurs et laissaient à la libre disposition le code source afin de lire et de modifier ce logiciel.

Un jour, le laboratoire achète une nouvelle imprimante. Mais le pilote de l'imprimante n'est pas fourni. Il n'y a donc plus aucune possibilité d'agir en cas de panne. Richard Stallman, qui travaillait dans ce laboratoire depuis 1971, entendit parler qu'un autre laboratoire possédait les sources du fameux pilote et les demanda. On lui répondit que le laboratoire s'était engagé à ne pas diffuser les sources du pilote. Particulièrement frustré, il décida de réagir en développant un système d'exploitation libre, autrement dit un système qui lui permettrait d'effectuer librement les modifications qu'il désirait et éventuellement de les communiquer à ceux qui en émettraient le souhait, comme il pouvait le faire auparavant. Stallman envisagea diverses formules pour son projet et finalement, afin de garantir le succès de ces logiciels, il choisira un système compatible avec Unix[3]. Son projet est nommé GNU selon une mode de l'époque, GNU étant un acronyme récursif qui signifie « GNU's Not Unix ». Afin de susciter de l'intérêt pour son projet, Stallman écrit un manifeste, *The GNU Manifesto* (Voir Annexe 1), dans lequel il explique ses motivations, ses objectifs. Compréhensible et direct, le style du manifeste parle aux développeurs qui adhèrent rapidement à son projet.

b) Le projet de Stallman

Une des premières préoccupations de Richard M. Stallman et des autres participants au projet GNU est de créer une association, la Free Software Foundation, qui permettra le développement de logiciels libres. La « Free » de Free Software Foundation

a Anecdote tirée du livre en ligne « Free as Freedom Richard Stallman's Crusade for Free Software » écrit par Sam Williams (Mars 2002) http://www.oreilly.com/openbook/freedom/ch01.html

1 Le code source est un ensemble d'instructions écrites dans un langage de programmation informatique de haut niveau, c'est-à-dire humainement compréhensible. Il faudra le compiler, c'est à dire le retranscrire en suite de 0 et de 1 pour qu'il puisse être interprété par un ordinateur.

2 un pilote (ou driver en anglais) permet à un système d'exploitation de reconnaître un certain matériel et de l'utiliser.

3 UNIX™ est le nom d'un système d'exploitation multitâche et multiutilisateur créé en 1969, à usage principalement professionnel, conceptuellement ouvert et fondé sur une approche par laquelle il offre de nombreux petits outils chacun dotés d'une mission spécifique. Il a donné naissance à une famille de systèmes, dont les plus populaires en 2007 sont Linux et Mac OS X.

doit être traduit par libre et non pas gratuit. Ces logiciels peuvent tout à fait être vendus, mais il existe toujours un moyen légal de se les procurer gratuitement.

L'autre préoccupation de Richard Stallman a été d'instaurer des conditions de distribution qui empêcheraient de transformer le logiciel GNU en logiciel propriétaire, puisque ce projet a comme volonté de construire un système informatique global, complet et libre, développé par un groupe social dont les membres ont partagé librement leurs travaux. Pour cela il créé la GNU General Public License (GPL). C'est un contrat qui lie le programmeur et l'utilisateur : l'utilisateur a le droit de copier, modifier, étudier, distribuer le programme, il a également accès au code source de celui-ci. En contrepartie, il s'engage à transmettre le programme, y compris les travaux dérivés ou si le programme est incorporé dans un autre, selon les mêmes termes et à garantir un accès au code source.

> *« In the GNU project, our aim is to give all users the freedom to redistribute and change GNU software. If middlemen could strip off the freedom, we might have many users, but those users would not have freedom. So instead of putting GNU software in the public domain, we « copyleft » it. Copyleft says anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. Copyleft guarantees that every user has freedom. »*[a]

Richard Stallman inventa l'idée du copyleft[1] en 1983, quand il démarra le projet GNU. Les premières ébauches de la GPL sont rudimentaires. Elles ne couvrent pas l'ensemble des besoins et ne protègent pas des utilisations illégitimes possibles. La version 1.0 apparaît en 1989. Il aura donc fallu plusieurs années à Richard Stallman et à la Free Software Foundation (FSF) pour concevoir une version satisfaisante. Elle sera remplacée par la version 2 en juin 1991. En 2005, Richard Stallman a commencé à écrire la version 3 de la GPL. En 2006, la Free Software Foundation a commencé une consultation du public d'une durée de 12 mois sur les possibles modifications de la GPL version 3. Le processus a été géré par la Free Software Foundation, la Software Freedom Law Center et la Free Software Foundation Europe. Finalement la troisième version de la GPL a été publiée le 29 juin 2007. Le premier projet à utiliser la nouvelle version de la licence est le logiciel d'archivage tar, version 1.18, publié par la FSF elle-même. Au 10 Mai 2008, 2281 projets avaient migré vers la GPL version 3 et 191 vers la LGPLv3.[b]

c) Rencontre avec Linus Torvalds

En 1990, le système GNU de Richard Stallman est presque terminé. Le seul composant qui manque est le noyau[2]. Or, en 1991, Linus Torvalds développe un noyau

a Extrait de l'article « What Is Copyleft » issue du site GNU Operating System : http://www.gnu.org/copyleft/copyleft.html

1 Le copyleft est la possibilité donnée par l'auteur d'un travail soumis au droit d'auteur (œuvre d'art, texte, programme informatique, etc.) de copier, d'utiliser, d'étudier, de modifier et de distribuer son œuvre dans la mesure où ces possibilités sont préservées.

b Source : http://gpl3.palamida.com

2 Un noyau de système d'exploitation (abrégé noyau, ou kernel en anglais), est la partie fondamentale de certains systèmes d'exploitation. Il gère les ressources de l'ordinateur et permet aux différents

compatible avec Unix, qui fût par la suite appelé Linux en honneur de son créateur et il décide de le mettre sous les termes de la GPL après quelques mois.

> *« A l'origine, Linux était simplement quelque chose que j'avais fait et le rendre disponible relevait plus d'un « regardez ce que j'ai fait » en espérant que quelqu'un le trouverait utile. Le rendre libre a permis à un plus grand nombre de développeurs d'y travailler et de l'étendre. Encore plus important a été le fait que d'un seul coup, le système a pu bénéficier des utilisateurs qui en testaient à la fois les bogues (erreurs) et l'utilisabilité. Aujourd'hui, il est évident que je profite de la renommée de Linux pour travailler, et je marchande donc ma réputation plus que Linux. »*[a]

C'est donc en 1992 que la rencontre entre Linux et le système GNU permet de fournir un système d'exploitation libre et complet, évitant l'utilisation du logiciel propriétaire d'Unix. Stallman était au M.I.T. à l'époque de la création du projet GNU et Linus Torvalds terminait ses études lorsqu'il a choisi de diffuser ce qui était son mémoire de fin d'études.

Le phénomène du logiciel libre et du copyleft a pu réellement prendre son envol avec l'apport de Linus Torvalds. GNU/Linux a également beaucoup profité de l'Internet balbutiant au début des années 90 et de son explosion au cours de ces dix dernières années.

Les logiciels libres sont maintenant développés par des milliers de personnes qui choisissent de les diffuser grâce à la General Public License. Les logiciels sont divers : applications de calcul, traitement de texte, messagerie, jeux, graphisme, musique et probablement tout ce qui peut être utile aux développeurs ou à leurs employeurs.

La facilité d'acquisition et de modification est un des facteurs essentiels à la propagation des logiciels libres. L'expression «Logiciel libre» fait référence à la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de liberté pour l'utilisateur du logiciel :

- La liberté d'exécuter le programme, pour tous les usages (liberté 0).
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins (liberté 1). Pour ceci l'accès au code source est une condition requise.
- La liberté de redistribuer des copies, donc d'aider votre voisin, (liberté 2).
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté (liberté 3). Pour ceci l'accès au code source est une condition requise.

Il est donc possible de redistribuer des copies, avec ou sans modification, gratuitement ou non, partout et à n'importe qui. Il n'y a pas d'obligation de demander ou de payer pour avoir le droit de le faire. Il est possible de faire des modifications et de les utiliser à titre personnel, pour le travail ou pour les loisirs. Si il y a des modifications, Il n'y a pas obligation de les déclarer, ou de prévenir quelqu'un de particulier.

composants - matériels et logiciels - de communiquer entre eux.

a Extrait traduit d'une interview de Linus Torvalds par « First Monday » un journal sur Internet (Mars 1998) : http://www.firstmonday.dk/issues/issue3_3/torvalds/

I.2 - Qu'est ce qui fait le succès du libre ?

Dans l'élan initié par le projet GNU, des milliers d'applications libres ont progressivement été développées, du plus petit programme, aux applications les plus pointues, et ce, dans tous les domaines. Certains projets ont été développés par des particuliers, d'autres par des sociétés commerciales mondialement connues souvent épaulées par une communauté active. Les libertés (accès au code source, possibilité d'améliorer le programme) offertes par les différentes licences libres - dont la GPL en particulier - ont grandement favorisé cet essor (reprises de projets abandonnés, forks[1], etc.).

En effet, le fait de pouvoir copier, modifier et diffuser librement les logiciels a permis de déployer une incroyable force de développement et ce grâce aussi à la croissance exponentielle d'Internet et à l'arrivée du haut débit. A travers Internet, des gens du monde entier peuvent participer, à distance, à la réalisation d'un projet commun, sans jamais se rencontrer. On peut donc imaginer la rapidité d'évolution de ces logiciels, et la réactivité des développeurs. Que l'on soit professionnel ou simple amateur, tout le monde peut participer, à sa manière, en fonction de ses connaissances et de son temps : participer à l'écriture, apporter de nouvelles idées sur les fonctionnalités, rédiger de la documentation, tester et alerter lorsqu'il y a des bugs[2] dans les programmes, aider les gens en difficulté sur les forums, etc.

a) Exemples de logiciels

Grâce à cette force, certains projets ont été projetés sur le devant de la scène, rattrapant et parfois dépassant leurs homologues propriétaires. Citons l'exemple de Firefox. Ce logiciel, en concurrence directe avec les Internet Explorer et Safari, a été créé par le Projet Mozilla, et développé par des milliers de volontaires dans le monde. Ce projet est soutenu et organisé par 3 associations :

Mozilla Foundation (aux Etats Unis) ; Mozilla Europe (à Paris) ; Mozilla Japan. Ces trois organisations bien qu'à but non lucratif, revendiquent 6 Millions d'utilisateurs en France pour Firefox et 85 Millions dans le monde. Tristan Nitot président de Mozilla Europe, déclare à titre de comparaison, que Firefox sort une nouvelle version environ tous les ans, tandis que pour Internet Explorer il s'est écoulé 5 ans entre les versions 6 et 7 « une éternité sur le web[a] ». De nombreuses innovations telles que l'utilisation des onglets, ou le correcteur orthographique intégré au navigateur, ont été popularisées par le Projet Mozilla.

Depuis Janvier 2005, en France, la gendarmerie a abandonné la fameuse

1 Fork est un mot anglais signifiant « fourche », « fourchette ». Il vient du mot latin furca (qui est également à l'origine de plusieurs mots en français comme bifurcation). C'est souvent dans ce sens figuré « d'embranchement », de « différentiation » qu'est employé cet anglicisme en informatique. Il désigne donc un objet (au sens large, cela peut être un projet) ayant une racine commune avec un second. Ces deux objets jumeaux au départ après s'être séparés suivent une évolution propre, différente.

2 Un bug ou bogue est une anomalie dans un programme informatique l'empêchant de fonctionner correctement. Sa gravité peut aller de bénigne (défauts d'affichage mineurs) à majeure (explosion du vol 501 de la fusée Ariane 5). Les bogues sont généralement dus à un problème de conception du logiciel.

a Source : Emission 8-Fi passée sur Direct-8. Cette vidéo est visible sur le site zdnet.fr à l'adresse suivante : http://www.zdnet.fr/partenaires/8-fi/0,3800004982,39367962,00.htm

suite Microsoft Office pour la remplacer par le logiciel libre Open Office[a]. Cette migration qui représente 80 000 postes, permet une économie de 2 millions d'euros par an, réalisée principalement sur les licences. Mais il s'agit aussi de simplifier la gestion de ces licences logicielles. En effet le réseau de la gendarmerie étant complexe, il est difficile d'un point de vue administratif de vérifier la conformité de toutes les licences déployées. Avec Open office, il n'y a plus de problème de licence : le logiciel est gratuit et téléchargeable selon deux licences : GPL (GNU General Public Licence) et LGPL (GNU Lesser General Public Licence). Par ailleurs, les ministères de l'Équipement, de l'Intérieur, de l'Économie, des Finances, de l'Industrie et l'Administration des douanes, sont aussi équipés de la suite Open Office.

b) Et plus largement :

Le « monde du libre », ne concerne pas que les logiciels. Il existe de nombreux projets basés sur le modèle de la GPL. Parmi ces projets, la Fondation Wikimedia est une organisation internationale à but non lucratif de droit américain, qui a pour but de soutenir le développement et la diffusion de contenu libre réutilisable par tout le monde, et ce de façon gratuite en plusieurs langues. Ses projets[b], qui sont parmi les plus grands succès dans le domaine de l'édition collaborative, utilisent la technologie du wiki[1] :

- *Wiktionary* est un projet visant à créer des dictionnaires et thésaurus libres dans chaque langue. Ce projet a démarré en décembre 2002 et est maintenant disponible dans plus de 100 langues avec plus de 1 500 000 entrées au total. La plus complète est la version anglophone, suivie par les versions francophone, vietnamienne et russe, chacune d'entre elles possédant plus de 100 000 entrées. Quarante autres langues possèdent au moins 1000 entrées.
- *Wikiquote* est un projet visant à constituer un répertoire de citations provenant de personnes célèbres, d'ouvrages, de discours, de films ou d'autres contenus intellectuellement intéressants. Wikiquote contient également des proverbes, des moyens mnémotechniques et des slogans. Le projet a été lancé en juillet 2003. En décembre 2006, il contenait plus de 50 000 pages dans plus de 30 langues. La plus complète est la version anglophone avec plus de 8000 pages. Les versions germanophone, polonaise et italienne possèdent chacune plus de 4000 articles.
- *Wikibooks* est un projet visant à constituer un fond de livres électroniques visant spécifiquement un public étudiant, comme des ouvrages, des manuels ou des livres tombés dans le domaine public et annotés. Ce projet est destiné à aider à la fois l'accès au savoir des étudiants et des professeurs du collège à l'université. La plus complète est la version anglophone avec plus de 22 000 chapitres. Le projet a été lancé en juillet 2003 et propose plus de 50 000 chapitres issus de 1000 ouvrages en 50 langues.

a Source : http://www.zdnet.fr/actualites/informatique/0,39040745,39203431,00.htm
b Source : http://wikimediafoundation.org/wiki/Nos_projets

1 Un wiki est un système de gestion de contenu de site web qui rend les pages web librement et également modifiables par tous les visiteurs autorisés. On utilise les wikis pour faciliter l'écriture collaborative de documents avec un minimum de contraintes.

- *Wikisource* est un projet multilingue lancé en novembre 2003 visant à construire un fond de textes, distribués sous la forme de contenu libre et ouvert. Ce n'est pas seulement un outil efficace permettant de stocker au format hypertexte des textes classiques, des textes de lois et d'autres textes libres de droits, il sert également de base de traduction de ces textes.
- *Wikispecies* est un projet visant à créer un répertoire central des espèces pour la taxonomie. Il est conçu spécifiquement pour répondre aux besoins des scientifiques et, en décembre 2007, contenait plus de 80 000 entrées.
- *Wikimedia Commons* est une médiathèque en ligne lancée en septembre 2004 visant à proposer un répertoire central de contenus libres (images, sons, vidéos, textes audio, etc.) pouvant être facilement utilisées par tous les projets Wikimedia. En novembre 2006, elle a atteint un million de fichiers multimédia.
- Le projet *Wikinews* a été lancé en décembre 2004 avec la mission de rapporter les actualités de façon large. En décembre 2006, Wikinews était actif dans 22 langues et avait produit au total plus de 25 000 articles d'actualité. Des contributeurs du monde entier rédigent ensemble des articles rassemblant des articles originaux et des compilations de sources externes. Toutes les nouvelles doivent être rédigées selon un point de vue neutre.
- *Wikipédia* est un projet visant à construire une encyclopédie libre dans chaque langue existant dans le monde. En pratique tout le monde est libre de contribuer. Le projet a démarré en janvier 2001 en une seule langue, l'anglais, et dépasse aujourd'hui les cinq millions d'articles dans plus de 200 langues. La plus grande Wikipédia est la version anglophone, avec plus d'un million et demi d'articles. Viennent ensuite les versions germanophone et francophone, avec chacune plus de 400 000 articles. Certaines Wikipédia diffusent ou prévoient de diffuser régulièrement des versions statiques de leur contenu. La Wikipédia germanophone est éditée deux fois par an sur DVD, en collaboration avec Directmedia Publishing.

Ces différents projets participent largement à la diffusion des connaissances, et favorisent l'accès à la culture pour tous. Malgré tout, se pose la question de la véracité des informations rencontrées sur ces différents sites. En effet, étant ouverts à tous, chacun peut créer des articles, et divulguer de fausses informations. Cependant, une étude[a] réalisée par « Nature[1] », a démontré que les grosses erreurs sont aux mêmes nombres que se soit dans l'encyclopédie professionnelle Britannica[2] ou dans l'encyclopédie virtuelle Wikipédia. De plus, comme dans l'ensemble des logiciels libres, chacun ayant la possibilité de visualiser et de modifier les articles, les contenus malveillants (informations

a Source : http://www.zdnet.fr/actualites/internet/0,39020774,39296098,00.htm

1 Nature est une, sinon la, revue scientifique généraliste de référence. C'est l'un des journaux scientifiques les plus anciens et les plus réputés au monde. Elle a été lancée en 1869 par le britannique Joseph Norman Lockyer

2 L'Encyclopædia Britannica est une encyclopédie en langue anglaise. D'origine britannique, elle est de nos jours éditée aux États-Unis.

erronées, virus informatiques[1] et logiciels espions[2]), sont immédiatement décelés et supprimés. Les logiciels libres ou open source sont sans doute beaucoup plus sécurisés que leurs homologues propriétaires.

c) Les distributions Linux

Parallèlement, de nombreuses distributions sont apparues suite à la rencontre du projet GNU et de Linux. Une distribution est le regroupement d'un système d'exploitation (ici Linux) et d'un ensemble de logiciels libres, allant de la bureautique, en passant par les applications serveur, les jeux et le multimédia. Le but de ces distributions est de répondre à différentes attentes :

- Proposer un choix de logiciels cohérent et stable (parmi la multitude existant).
- Proposer un système de gestion de ces logiciels qui permette l'installation, la mise à jour et l'application des correctifs et autres patchs[3] de sécurité.
- Proposer éventuellement un ensemble d'outils d'administration (installateur, reconnaissance du matériel, etc.)

Les distributions GNU/Linux (il en existe plus d'une centaine listée par le site internet http://distrowatch.com) sont de véritables systèmes « clé en main » tout comme Microsoft ou Mac le proposent. Certains projets ont en effet le désir de faciliter la mise en œuvre et de réaliser des distributions réunissant stabilité et convivialité. Elles s'adressent aussi bien aux particuliers qu'aux professionnels, débutants ou confirmés, qui souhaitent disposer d'un système d'exploitation libre et sécurisé.

Une des distributions la plus prometteuse est développée par l'entreprise Canonical : Ubuntu Linux. Canonical est une organisation basée en Europe, chargée de développer, distribuer et de fournir du support pour les logiciels open source et la communauté. Ubuntu est un projet développé et soutenu par la communauté. Depuis son lancement en Octobre 2004, Ubuntu est devenu une des distributions les plus utilisée, avec des millions d'utilisateurs dans le monde.

> *« Ubuntu sera toujours gratuit et libre en téléchargement, utilisation, distribution. Avec ces objectifs en tête, Ubuntu a pour but de devenir le système Linux le plus largement utilisé, il est au centre d'un écosystème global de logiciels libre. »*[a]

On apprenait en Mars 2007 que l'Assemblée nationale, après avoir lancé un

1 Au sens strict, un virus informatique est un programme informatique écrit dans le but de se propager à d'autres ordinateurs. Il peut aussi avoir comme effet, recherché ou non, de nuire en perturbant plus ou moins gravement le fonctionnement de l'ordinateur infecté. Au sens large, on utilise souvent et abusivement le mot virus pour désigner toute forme de programme malveillant (malware).

2 Un logiciel espion (espiogiciel, mouchard ou en anglais spyware) est un logiciel malveillant qui s'installe dans un ordinateur dans le but de collecter et transférer des informations sur l'environnement dans lequel il s'est installé, très souvent sans que l'utilisateur n'en ait connaissance.

3 Un patch est une section de code que l'on ajoute à un logiciel, pour y apporter des modifications mineures : correction d'un bogue, traduction, crack.

a Traduit à partir du site de Ubuntu : http://www.ubuntu.com

appel d'offres en début d'année, pour remplacer son parc informatique qui datait de 2002, avait penché pour la distribution Ubuntu. Ce sont 1154 postes informatiques qui sont équipés de logiciels libres depuis Juin 2007[a] : Open Office pour la bureautique, Firefox comme navigateur Internet, et Thunderbird pour la messagerie. Cette décision a été prise en partie par souci économique. Mais les logiciels libres ou open source sont aussi un gage de transparence parce qu'il est possible de vérifier chaque ligne de code, ce qui limite le danger des failles de sécurité et l'intrusion de petits programmes malveillants, grâce à la vigilance de la communauté de développeurs.

Aujourd'hui, le logiciel libre est clairement sorti de la sphère des « hackers[1] ». Ce modèle de développement de logiciels est soutenu et utilisé par des institutions éducatives, industrielles (comme NeXT, Intel, Motorola, certaines écoles en Belgique) et gouvernementales (notamment la NASA et l'U.S. Air Force, mais aussi dans les administrations belges, qui ont pour la plupart installé des serveurs « pare-feu[2] » sous GNU/Linux). Le modèle de GNU/Linux a connu un tel succès, que des sociétés commerciales ont adopté cette technique de développement. Dans le monde informatique, il est devenu impossible de négliger les logiciels libres. C'est donc une des raisons pour laquelle je m'y intéresse. Les logiciels libres sont en vogue. Proposer une formation M.A.O. utilisant exclusivement Linux me semble plus judicieux, qu'avec des logiciels propriétaires dont les stages sont déjà très nombreux.

Le chapitre suivant traitera des différentes licences libres. J'étofferai mon propos en citant d'autres licences utilisées plus généralement dans l'art. En effet lorsque l'on travaille sous Linux, que l'on créer de la musique, des vidéos, des textes et que l'on adopte cette philosophie de l'Open Source, on veut sans doute partager son travail dans ces mêmes conditions. Sans aborder toutes les possibilités, car il y a pléthore, je présenterai ici les licences les plus répandues.

a Sources : http://www.clubic.com/actualite-65682-assemblee-nationale-equipe-logiciels-libres.html et http://www.pcinpact.com/actu/news/35163-assemblee-nationale-ubuntu-linagora-unilog-m.htm

1 Il existe plusieurs définitions pour désigner un **hacker**. Ici le hacker désigne un virtuose pouvant intervenir dans différents domaines comme la programmation, l'architecture matérielle d'un ordinateur, l'administration système, l'administration réseau, la sécurité informatique ou tout autre domaine. A ne pas confondre avec le hacker disposant d'un savoir-faire exceptionnel dans la maîtrise de la sécurité informatique et donc des moyens de déjouer cette sécurité.

2 Un pare-feu (firewall en anglais) est un élément du réseau informatique, logiciel et/ou matériel, qui a pour fonction de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communication autorisés ou interdits.

II. Les licences : Explication / Exemples

II.1 - Pour le logiciel

a) État des lieux.

Généralement, une œuvre est automatiquement protégée par le droit d'auteur ou le copyright pour les pays anglo-saxons. Cette protection est bénéfique à l'auteur et interdit toute forme d'utilisation par des tiers. Si l'on désire mettre une œuvre (logiciel ou autre) à disposition de tous, et notamment la diffuser sur Internet, il faut lui adjoindre une licence, qui exprimera, explicitement notre volonté de la partager. Cette licence, permet de définir les droits et les obligations éventuelles imposés aux utilisateurs. Une licence est dite libre, lorsqu'elle garantie un certain nombre de libertés à l'utilisateur de l'œuvre associée.

Le succès des logiciels libres a créé des tensions au sein de la communauté. C'est ainsi qu'à la fin des années 90, le mouvement s'est scindé en deux groupes : La Free Software Foundation créée par Richard Stallman ; l'Open Source Initiative, créée en 1998 par Eric Raymond et Bruce Perens.

- La FSF considérée comme radicale, adopte l'idéologie suivante : *« du libre ne peut s'appuyer que sur du libre et ne déboucher que sur du libre »*. Elle a rédigé plusieurs licences dont la GNU GPL (GNU General Public Licence) et la GNU LGPL (GNU Lesser General Public Licence).
- L'Open Source Initiative (OSI), plus modérée, accepte le mélange entre les logiciels libres et propriétaires : « un logiciel libre peut s'appuyer sur un logiciel propriétaire et/ou donner autre chose que du logiciel libre ». La Mozilla Public Licence (exemple : Firefox), fonctionne sur ce principe. L'Open Source Initiative n'a pas vraiment rédigé de texte mais a créé un système de labellisation et se limite à recenser les licences, satisfaisant ainsi leurs principes et les projets qui en découlent : environ 40 licences différentes et plus de 60 000 projets.

Il existe de nombreuses licences, s'inspirant de ces deux mouvements. Il faut savoir que les dénominations « libre », « open source » ou « copyleft » n'ont pas de valeur juridique. Il faut aussi rappeler que libre ne signifie pas forcément gratuit. Voici quelques différences :

- Avec une licence libre (ou open source) l'auteur autorise sans exclusivité ni discrimination, la copie, la diffusion et la modification de l'œuvre, en exigeant que les sources ou références de l'œuvre soient mentionnées et que celle-ci soit rendue accessible. Cependant, avec une licence open source, l'auteur d'une version modifiée peut placer sa propre contribution sous une licence plus restrictive ; de ce fait, l'éventuel enrichissement qu'il aura apporté à l'œuvre ne profitera pas de la même façon à la communauté puisqu'il ne sera pas permis à d'autres de modifier cette nouvelle œuvre à leur tour.
- Une licence copyleft est une licence libre qui ajoute une clause de pérennité. Elle fait obligation aux auteurs de versions modifiées de perpétuer les mêmes libertés sur leurs contributions (via la même licence ou une licence équivalente).

- Les licences semi-libres sont celles qui apportent des restrictions partielles à la diffusion ou aux possibilités de modification de l'œuvre. Les plus restrictives autorisent au moins l'accès libre aux données ou leur téléchargement pour un usage privé. Dans ce cas, la mise à disposition suffit, une licence n'est pas nécessaire.

Historiquement, la plupart des licences libres s'inspirent ou se réclament de la GNU GPL créée par Richard Stallman en 1983 pour les logiciels, puis fixée par la Free Software Foundation en 1989. Selon leur degré d'attachement aux principes fondamentaux du copyleft, ces licences en font des adaptations qui peuvent varier en fonction du champ d'application de la licence mais aussi des usages qui seront autorisés ou non.

b) La GNU GPL : GNU General Public Licence.

Le But de la GNU GPL est d'encadrer juridiquement des faits techniques : code source, application, traduction, etc. et s'applique exclusivement à des programmes informatiques. La GPL est un complément de la mention copyright ou du droit d'auteur : elle précise l'usage, les conditions de distribution du programme en question. En effet, le copyright est explicitement cité dans le préambule :

> *« Les développeurs qui utilisent la GPL GNU protègent vos droits en deux étapes : (1) ils affirment leur droits d'auteur ("copyright") sur le logiciel, et (2) vous accordent cette Licence qui vous donne la permission légale de le copier, le distribuer et/ou le modifier. »*[a]

Comme indiqué plus haut, la GNU GPL est créée pour défendre le logiciel libre, celui-ci se définissant, selon la Free Software Foundation, suivant quatre « libertés » :

- La liberté d'exécuter le programme, pour tous les usages;
- La liberté d'étudier le fonctionnement du programme et de l'adapter à vos besoins. Pour ceci, l'accès au code source est une condition requise;
- La liberté de redistribuer des copies, donc d'aider votre voisin;
- La liberté d'améliorer le programme et de publier vos améliorations pour en faire profiter toute la communauté. Pour ceci, l'accès au code source est une condition requise.

L'idée de Richard Stallman a été de concevoir une licence protégeant les logiciels libres contre une appropriation par qui que ce soit : lorsqu'un logiciel est régit par la GPL, il reste libre, même s'il est intégré dans un autre programme.

Richard Stallman conçoit la GPL dans le but explicite de défendre les logiciels libres d'une appropriation par qui que ce soit. Il y a dans le projet GNU la volonté d'un système libre dont les composantes ne sont pas « propriétaires » et ne le deviennent pas. Autrement dit, un logiciel lorsqu'il devient GPL, le reste, même s'il est intégré dans un autre programme.

---

a Sources : http://www.gnu.org/licenses/gpl.html. On peut lire la traduction française non officielle à cette adresse : http://www.rodage.org/gpl-3.0.fr.html

« [Vous devez] Distribuer sous les termes de la GPL l'ensemble de toute réalisation contenant tout ou partie du programme [sous GPL], avec ou sans modification. »[a]

L'article 1 de la GPL consacre la possibilité pour l'utilisateur de copier et distribuer des copies conformes du programme à condition de placer sur chaque copie le copyright approprié, de ne pas modifier ou omettre toutes les stipulations se référant à la présente Licence.

Donc, la licence GPL permet à l'utilisateur de jouir des droits de copie, de distribution et de modification et le force à les transmettre à son tour, y compris dans les travaux dérivés ou dans le cas d'une intégration à un autre programme, ces derniers devant être également sous GPL. Dès lors, il est impossible d'inclure un programme sous copyleft dans un programme propriétaire.

c) La GNU LGPL (Lesser General Public Licence)

Cette licence est une version dérivée de la GPL. Elle a été créée afin d'assouplir certaines conditions de la licence GPL qui s'avère parfois trop restrictive. Cette licence donne le droit d'intégrer des bibliothèques[1] libres dans des programmes ou logiciels qui ne le sont pas forcément. Un auteur peut donc utiliser ou mettre à disposition une bibliothèque en la rendant libre, même s'il développe un logiciel propriétaire.

II.2 - Au delà du logiciel :

Bien sûr, d'autres licences ont vu le jour et s'inspirant de la philosophie de la Free Software Foundation, elles ont élargie leur champ d'application, à l'artistique, à l'éducation, à la documentation. D'une manière générale, toutes les licences du copyleft, reposent sur le copyright. Il s'agit toujours d'un contrat par lequel l'auteur, titulaire des droits d'une œuvre, autorise une personne à effectuer des actions (modifications, copies, diffusions, etc.) qui autrement lui seraient interdites. Les créations qui sont protégées par les droits d'auteur peuvent potentiellement se faire « copylefter ». Autrement dit, toutes les œuvres d'art et de littératures : musique, émission de télévision, livres scolaires, poèmes ou sites web. Face à la diversité du type de production, les licences contiennent des termes spécifiques propres aux moyens de diffusion ou de reproduction susceptible d'être utilisés. Au vu du nombre de licences différentes, je vais en citer quelques unes, qui me paraissent importantes.

---

a Article 2 b de la GPL : http://www.linux-france.org/article/these/gpl.html

1 En informatique, une bibliothèque ou librairie logicielle est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. L'intérêt des bibliothèques réside dans le fait qu'elles contiennent du code utile que l'on ne désire pas avoir à réécrire à chaque fois.

a) GNU Free Documentation Licence :

La FDL a été conçue par la Free Software Foundation en 2000, pour la documentation des logiciels libres sous Gnu GPL, mais elle peut également convenir à d'autres documents, en particulier pour des manuels d'enseignement. C'est une licence libre et copyleft qui se présente comme un manuel d'utilisation extrêmement détaillé de la licence. Elle garantit la liberté d'utilisation, à savoir : de copier ou de redistribuer, avec ou sans modification, commercialement ou non. La FSF recommande cette licence pour les manuels techniques, mais également pour « *les travaux destinés à des fins d'enseignements ou devant servir de documents de référence*[1] ». Cette licence à quelques particularités :

- Elle règle les problèmes concernant les droits de paternité, garantissant ainsi la reconnaissance du travail de l'auteur, par rapport aux versions modifiées par des tiers.
- On ne peut utiliser la FDL que sur des documents qui sont numériques et « transparents ». Cela signifie que les formats de fichiers doivent être publics, lisibles par tous et rendre le texte techniquement modifiable. Toute copie « opaque », tel qu'une copie papier, ou un fichier PDF, doit indiquer l'URL d'un exemplaire transparent.
- La FDL autorise la protection de certaines parties du texte contre les modifications : toutes les parties non techniques, c'est à dire celles ne devant pas être mises à jour dans le cadre de l'évolution de l'informatique.

Richard Stallman écrivait dans un article daté de Mars 2000, que la FDL est aussi destinée à des éditeurs commerciaux, il espérait que des rédacteurs de documents techniques puissent être rémunérés :

> *« La FDL de GNU, ou GFDL, se veut un moyen d'inciter des éditeurs commerciaux à financer de la documentation libre sans pour autant renoncer à leurs libertés les plus vitales. Le «texte de couverture», ainsi que certains autres aspects de la licence concernant la couverture, la page de titre, l'historique et les cautions, y ont été placés afin de rendre la licence attirante pour des éditeurs commerciaux et pour des livres dont les auteurs sont payés. »*[a]

b) Licence Art Libre (LAL) :

La Licence Art Libre a été élaborée en juillet 2000 par Copyleft_Attitude, une association composée d'artistes, de juristes et d'informaticiens, au terme de 7 mois de colloques, workshops, réunions, et discussions en ligne. Elle a été révisée en novembre 2003.

> *« "Copyleft Attitude" a pour objectif de faire connaître et promouvoir la notion de copyleft dans le domaine de l'art et au delà. Prendre modèle sur les pratiques liées aux logiciels libres pour s'en inspirer*

1 Citation du préambule de la GNU Free Documentation Licence.

a Extrait d'un article visible à cette adresse : http://www.gnu.org/licenses/why-gfdl.fr.html

et les adapter à la création hors logiciel. C'est la raison pour laquelle nous avons mis au point la Licence Art Libre. »[a]

Cette licence autorise sans restriction, la copie, la modification et la diffusion gratuite ou payante d'une œuvre. En revanche, elle donne l'obligation de transmettre l'œuvre modifiée avec la même licence, le nom de tous les auteurs ayant participé (versions originales et modifiées si elles existent) ainsi que les références permettant d'accéder au travail. Il est aussi possible d'intégrer l'œuvre dans un ensemble tant qu'il n'y a pas de contradiction avec les dispositions de la licence. En d'autres termes, l'œuvre doit rester extractible afin que les libertés conférées par la LAL ne soient pas altérées.

- La LAL introduit la notion d'exemplaire original : la référence. Cette licence permet donc, dans le cas d'œuvre autographe de préserver cet exemplaire original et n'autorise les modifications que sur les copies. En revanche, dans le cas d'autres œuvres, cette règle devient artificielle donc indifférente.
- La LAL définit comme œuvre commune, l'ensemble des œuvres liées entre elles par l'usage du droit de copie, de modification ou incorporation.
- La LAL est rédigée en français et se distingue de bien des licences par la rigueur et la concision de son écriture. Formulée en termes à la fois généraux et précis, elle évite les énumérations ou les recommandations techniques trop détaillées, ce qui permet d'avoir une certaine souplesse d'interprétation quant à ses modalités d'application. Elle est lisible par les non juristes. En outre, une très bonne FAQ[1] y est associée.
- La LAL spécifie être soumise à la législation française ; le fait de s'adosser à une loi nationale rend l'interprétation de la licence beaucoup plus fiable, y compris dans le cadre d'une application internationale. En effet, dès lors qu'un contrat ne concerne pas des consommateurs, la loi applicable sera de préférence celle mentionnée dans le contrat (loi du pays *pour* laquelle la protection est réclamée), et le cas échéant, la loi locale (loi du pays *dans* laquelle la protection est réclamée.
- La Licence Art Libre est recommandée par la Free Software Foundation pour les œuvres non-logicielles.

c) Creative Common :

L'organisation Creative Commons a été fondée en 2001 aux États Unis par un groupe de juristes et d'experts en informatique. Les licences Creative Commons ont été rédigées en langue anglaise et en référence à la législation américaine du copyright. Ce n'est qu'en 2004, que les licences Creative Commons ont été adaptées aux législations d'autres pays. En France, la traduction, et la réécriture ont été effectuées par le groupe de recherche IDL (Informatique, Droit et Linguistique) du CERSA[2]. La version 1.0 de la licence a été mise en ligne le 2 décembre 2003. Suite à des discussions ouvertes entre les

a Source et site Officiel : http://artlibre.org/

1 Une foire aux questions, par rétro-acronymie à partir de l'acronyme anglais FAQ pour Frequently Asked Questions (« questions fréquemment posées »), est une liste faisant la synthèse des questions posées de manière récurrente sur un sujet donné, accompagnées des réponses correspondantes.

2 **CERSA** : Centre d'études et de recherches de science administrative

différents membres de la communauté (juristes, chercheurs, auteurs,...), une version 2.0 a été lancée le 19 Novembre 2004. Actuellement, aux États-unis il existe une version 2.5, et en France, la version 3.0 est en discussion.

Creative Commons[a] propose 6 contrats-types pour la mise à disposition d'œuvres sur Internet, résultant d'une combinaison d'options modulables. Ces contrats ont pour but de facilité l'utilisation d'œuvres (textes, photos, musique, site Internet,...) par des tiers. En effet, toute oeuvre est protégée par les droits d'auteur. Ces droits interdisent la reproduction, la diffusion, ou la modification. Les licences Creative Commons permettent d'expliciter la volonté de l'auteur de partager son oeuvre et d'informer le public qu'une utilisation est consentie à titre gracieux, mais selon certaines conditions.

La facilité d'utilisation de ces licences en kit a assuré un très grand succès à Creative Commons dont on voit le logo un peu partout sur les pages Web. Il faut tout de même être attentif aux risques de confusion entre ces licences, qui se signalent par des logos identiques, mais dont les implications juridiques sont bien différentes les unes des autres.

Voici un tableau des différentes possibilités :

Paternité			
Paternité ; Pas de modifications			
Paternité ; Pas d'Utilisation Commerciale ; Pas de Modification			
Paternité ; Pas d'Utilisation Commerciale			
Paternité ; Pas d'Utilisation Commerciale ; Partage des Conditions Initiales à l'Identique			
Paternité ; Partage des Conditions Initiales à l'Identique.minicomputer			

Signification de chaque option :

- Paternité : l'œuvre peut être librement utilisée, à la condition de l'attribuer à son l'auteur en citant son nom.
- Pas d'Utilisation Commerciale : le titulaire de droits peut autoriser tous les types d'utilisation ou au contraire restreindre aux utilisations non commerciales (les utilisations commerciales restant soumises à son autorisation).
- Pas de Modification : le titulaire de droits peut continuer à réserver la faculté de réaliser des œuvres dérivées ou au contraire autoriser à l'avance les modifications, traductions...
- Partage à l'identique des Conditions Initiales : à la possibilité d'autoriser à

a Source et site officiel des Creative Commons – France : http://fr.creativecommons.org/

l'avance les modifications peut se superposer l'obligation pour les œuvres dites dérivées d'être proposées au public avec les mêmes libertés (sous les mêmes options Creative Commons) que l'œuvre originaire.

D'autres options sont disponibles selon la législation américaine mais n'ont pas encore été transcrites en droit français. Ces options sont adaptées aux besoins du sampling[1], des pays en voie de développement, du partage de la musique...

d) Petit bémol

L'apparition de licences telles que les Creatives Commons, ou la Licence Art Libre basées sur la philosophie initiée par la Free Software Foundation et ses différentes licences GNU, a permis la création de nouveaux médias de diffusion et de nouvelles perspectives de collaboration artistique, notamment grâce au droit de modification. Tout de même, une remarque importante, évoquée sur le site Internet de la communauté Art Libre, concerne l'impossibilité pour les artistes sociétaires de la SACEM de mettre leurs œuvres sous copyleft.

> *« [...] Tout auteur, auteur-réalisateur, ou compositeur admis à adhérer aux présents Statuts fait apport à la société, du fait même de cette adhésion, en tous pays et pour la durée de la société, du droit d'autoriser ou d'interdire l'exécution ou la représentation publique de ses œuvres, dès que créées. -- Statuts de la SACEM, Article premier. »*

> *« Par cet acte d'adhésion, il s'engage notamment : [...] - de ne conclure aucune convention qui dispose au profit de qui que ce soit des droits dont il a fait apport à la société; [...] » Règlement général de la SACEM, Chapitre 5 - Règles communes à tous les membres de la société ; I. Devoirs généraux, Article 29. »*

1 En musique, l'échantillonnage ou sampling, est la numérisation de documents sonores ou échantillons (samples), effectuée à intervalles réguliers à l'aide d'un échantillonneur (ou sampler).

III. Le Libre et l'Audio

Après avoir présenté les logiciels libres, leur philosophie, puis certaines licences disponibles, je vais maintenant parler des outils qui existent et qui me permettront de créer une formation M.A.O. Dans une première partie, je justifierai le choix des logiciels libres. Puis je présenterai ceux qui me semblent les plus aboutis et qui ont un intérêt particulier pour la création musicale. Je pourrai donc envisager de les inclure dans la formation.

III.1 - Pourquoi utiliser des logiciels libres ?

a) Éthique

Tout d'abord, utiliser des Logiciels Libres est une démarche éthique : c'est la volonté d'utiliser des logiciels réalisés avec l'objectif de créer un bien commun dans l'intérêt général et non pas des logiciels créés pour servir des intérêts privés. Par exemple, l'idée de partage des connaissances véhiculée par la philosophie du Logiciel Libre est en adéquation avec les valeurs de l'enseignement public. L'utilisation de tels logiciels dans une formation serait donc cohérente.

b) Libre accès

Les Logiciels Libres sont librement accessibles, copiables et diffusables. Leur utilisation permet donc un accès moins onéreux aux technologies de l'information. Leur coût, généralement plus faible que celui des logiciels propriétaires, permet d'engendrer des économies sur l'achat des licences. Les économies ainsi réalisées peuvent par exemple permettre d'investir dans des développements pour améliorer les Logiciels Libres utilisés, dans la formation, dans le support technique ou du nouveau matériel. Ces investissements peuvent ainsi financer des sociétés de services locales plutôt que des éditeurs de logiciels situés la plupart du temps à l'étranger. Pour le responsable informatique d'une grande structure, la gestion des licences est simplifiée avec le Logiciel Libre : on peut le copier sans compter.

c) Indépendance et pérennité

La disponibilité du code source des Logiciels Libres permet d'être indépendant du fournisseur de logiciel. Ceci a plusieurs avantages :

- si le fournisseur de logiciels disparaît ou change son offre, il est possible de faire appel à un autre fournisseur, en conservant la solution technique actuelle;
- il est possible de modifier soi-même le code source du logiciel, ou d'employer des développeurs, afin d'ajouter des fonctionnalités ou de corriger des erreurs ;

D'autre part, les Logiciels Libres utilisent en général des formats et des protocoles ouverts, dont le fonctionnement est publiquement connu. Leur usage permet à

l'utilisateur de ne pas être dépendant d'un logiciel particulier : il peut en changer librement.

d) Qualités techniques

La disponibilité du code source permet à des milliers de développeurs de vérifier en permanence ce code source, améliorant ainsi la fiabilité et la sécurité des Logiciels Libres. En particulier, la communauté du Logiciel Libre est très réactive en ce qui concerne la correction des problèmes de sécurité par rapport aux éditeurs de logiciels propriétaires.

Les Logiciels Libres sont souvent indépendants des contraintes commerciales : le développement n'est pas dirigé par la rentabilité et les corrections de problèmes mineurs ou l'internationalisation (traduction dans plusieurs langues), ne sont donc pas négligées. Pour beaucoup de ces logiciels, il n'y a pas d'impératifs de calendrier : les nouvelles versions sortent lorsqu'elles ont un atteint un niveau de qualité suffisant.

e) Communauté du Logiciel Libre

Autour de l'idée des logiciels libres s'est constituée une importante communauté de programmeurs, d'utilisateurs, de traducteurs, de graphistes. Ceux-ci travaillent dans des dizaines de milliers de projets, dont l'envergure et le mode d'organisation sont très variables. Cependant, dans la majorité des cas, le développement est massivement décentralisé et Internet est au cœur de l'organisation des projets. C'est grâce au courrier électronique, aux listes de discussions et à tous les autres outils qu'offre Internet que le Logiciel Libre peut prospérer. Il est intéressant de noter qu'Internet existe aussi grâce aux logiciels libres (Ainsi, 70% des sites Web utilisent le logiciel libre Apache).

Dans ce modèle communautaire de développement, la relation entre l'utilisateur et le développeur n'est plus une relation de client à fournisseur, mais de personne à personne, privilégiant l'entraide. L'utilisateur devient acteur du processus de création, et participe à l'amélioration du logiciel. Les Logiciels Libres disposent donc d'une importante base d'utilisateurs-testeurs, avec laquelle les éditeurs de logiciels propriétaires peuvent difficilement rivaliser.

Les logiciels libres sont réalisés par des milliers de développeurs à travers le monde. Ces personnes travaillent pour des entreprises, des administrations publiques ou territoriales, des laboratoires de recherche publics ou privés, ou sont de simples programmeurs passionnés, férus d'informatique.

f) Éviter l'utilisation de formats fermés.

Un format de fichier décrit la manière dont sont stockées les informations. En général, les logiciels libres utilisent des formats et des protocoles ouverts, c'est-à-dire des formats de fichiers ou des protocoles de communication dont les spécifications sont librement accessibles. Même lorsque vous utilisez des applications propriétaires, il est important d'utiliser des normes ou des standards ouverts car leur utilisation offre plusieurs avantages :

- elle permet de garantir la pérennité des informations. En effet, que se passerait-il si dans 10 ans on ne pouvait plus lire des documents importants parce que la société qui éditait le logiciel a fait faillite ou que le format de fichier est devenu obsolète ?
- elle permet également de garantir l'interopérabilité, c'est-à-dire la capacité à travailler avec des systèmes et des logiciels différents, au choix de l'utilisateur. En utilisant un standard ouvert, vous pouvez envoyer un document à un correspondant sans le contraindre à utiliser un logiciel particulier pour le lire.
- l'indépendance technique est garantie, l'utilisateur n'est pas enfermé dans l'utilisation d'un logiciel en particulier et peut en changer librement.

g) Les limites des logiciels libres ?

En tant qu'utilisateur quotidien de Linux et des logiciels libres, il est difficile de porter un regard objectif sur les avantages et les inconvénients d'un tel système. Je peux tout de même souligner certains points. Que l'on soit amateur ou bien professionnel, musicien, graphiste ou développeur, chaque personne aura une utilisation spécifique de son outil informatique, avec des besoins différents. On ne peut donc pas dire que les logiciels libres "c'est mieux" car ce n'est pas toujours vrai. Bien que par exemple dans le domaine de l'hébergement Web et des serveurs on compte environ 70%[a] des sites Internet qui utilisent des serveurs Apache (une application libre dont la fiabilité à été prouvée depuis longtemps) ; dans d'autres domaines, tels que le graphisme et la production audiovisuelle certain logiciels manquent de maturité et n'ont pas toutes les fonctions requises pour être considérés ou en tout cas utilisés comme des applications professionnelles.

Bien que certains projets comme Firefox ou Open Office, réussissent à fédérer de nombreux contributeurs à travers le monde, certains projets restent des loisirs et finissent par être abandonnés faute de temps. En effet, de nombreux projets sont menés par des développeurs sur leur temps libre et il est difficile de faire durer un projet si l'on est seul à le maintenir. Tout de même si la demande est importante de la part d'utilisateurs, quelqu'un peut tout à fait reprendre le projet et continuer à le développer, car la licence le permet. Mais se posera le problème de la compréhension du code. Chaque développeur a sa propre logique, car il n'y a jamais une seule manière de faire et les commentaires à l'intérieur des programmes sont souvent rares. Il peut donc être difficile de s'approprier le programme. D'un autre côté, plus il y aura de monde à adopter les logiciels libres, plus il y aura de développeurs potentiels pour se joindre à des projets existants, reprendre ceux qui ne sont plus maintenus ou en créer de nouveaux.

À ce titre, on peut lire de nombreux articles sur les logiciels libres et sur la volonté de certaines organisations telles que l'AFUL[1], l'APRIL[2], ou encore le pôle de compétence des logiciels libres du SCÉRÉN[3] coordonné par Jean-Pierre Archambault, de

---

a Source : http://www.etrangers.lautre.net/04-03/icones.htm

1 L'Association francophone des utilisateurs de Linux et des logiciels libres (AFUL) a pour principal objectif de promouvoir, directement ou indirectement, les logiciels libres et notamment les systèmes d'exploitation libres comme GNU-Linux, ainsi que l'usage des standards ouverts.

2 L'« Association pour la Promotion et la Recherche en Informatique Libre » (APRIL) a été fondée en novembre 1996 afin de faire la promotion du logiciel libre en France, et aussi dans l'espace francophone.

3 Le « SCÉRÉN » (Services, culture, éditions, ressources pour l'Éducation nationale), est placé sous la tutelle du ministère de l'Éducation nationale. Le réseau SCÉRÉN s'articule autour du Centre national de

démocratiser leur utilisation. Ce dernier, professeur agrégé de mathématiques, chargé de mission veille technologique au CNDP-CRDP de Paris, où il assure entre autre la responsabilité du dossier des logiciels libres, défend notamment leur utilisation dans l'éducation nationale. Parmi les intérêts d'introduire les logiciels libres dans l'éducation je citerai les raisons économiques d'une part et l'absence de toute discrimination d'autre part. En effet, quel élève, collégien, lycéen ou étudiant peut acheter une licence de la suite Office de chez Microsoft ? Quel élève peut acheter une licence pour Photoshop, Flash ou Première de chez Adobe ? A la fin de son cours, l'enseignant demandera sûrement de finir l'exercice commencé à la maison. Cela incitera les élèves au piratage[1] des logiciels utilisés à l'école, chose qui ne peut arriver avec l'utilisation de logiciels libres puisque la diffusion et la copie sont autorisées.

La limite dans ces propos, et il faut être réaliste, c'est le décalage qu'il peut y avoir avec le monde professionnel. Car le but d'une formation c'est de préparer ses élèves à l'insertion professionnelle et souvent, les entreprises utilisent des logiciels propriétaires (Pro-Tools, Final Cut, etc.). Il est donc important et nécessaire de former les étudiants aux "standards" et de leur donner un maximum de chance pour la suite. Avec le temps, les lacunes actuelles des logiciels libres seront sans aucun doute comblées et les professionnels commenceront à y regarder de plus près. C'est pour cela qu'il faut s'y préparer et s'y intéresser dès maintenant.

III.2 - Un système d'exploitation libre

Avec le développement des licences et des logiciels libres, on a vu apparaître des alternatives aux solutions propriétaires. Même si certains logiciels libres manquent de maturité, d'autres sont plutôt prometteurs. Dans tous les cas, il semble que Linux soit plus souple et plus modulaire. En effet, il existe peu de logiciels "multifonctions", chacun effectue une tâche bien précise (enregistreur, séquenceur midi, synthétiseur, etc...). Il faut utiliser plusieurs logiciels que l'on interconnectera à l'aide du serveur JACK, que je présenterai plus loin.

Pour faire de la musique "librement", plusieurs solutions s'offrent à nous. Il existe un certain nombre de logiciels libres multi-plateformes, c'est à dire qu'ils fonctionnent aussi bien sous Linux que sous Windows ou MAC. Mais le mieux est de se tourner vers une distribution entièrement libre où le choix de logiciel sera beaucoup plus évident. Il existe pour cela des distributions dédiées et optimisées pour l'Audio. Je présenterai dans un premier temps les distributions les plus connues ainsi que les particularités du système Linux pour la création musicale. Puis je parlerai des logiciels.

documentation pédagogique (CNDP) qui pilote 31 centres régionaux, eux-mêmes chargés d'animer les centres départementaux et locaux

1 Un pirate informatique est une personne commettant des actes considérés comme des délits ou des crimes dont l'objet ou l'arme est lié à l'informatique.

a) Les bases de l'audio sous Linux

Différentes distributions :

De nombreuses distributions se sont déclinées et spécialisées dans divers domaines : bureautique, serveurs, multimédia et même Linux USB (pour transporter son système d'exploitation sur sa clé USB). Sans parler des Slackware ou Gentoo qui sont très complexes pour un utilisateur inexpérimenté, de nombreuses distributions sont dérivées de RedHat et de Debian.

RedHat est une des distributions les plus populaires. Aujourd'hui commerciale (et axée sur les entreprises), elle a cédé une branche Linux communautaire appelée Fedora. Elle utilise un système de paquets[1] binaires .rpm.

Debian est une distribution très performante. Complètement libre, très bien structurée, à développement exclusivement communautaire. Nettement plus difficile à installer qu'une Mandriva (basée sur RedHat), la distribution Debian possède un système de gestion de paquets largement supérieur. Cette distribution est connue pour son niveau technique d'excellence et sa stabilité. Elle utilise un système de paquets binaires .deb.

Pour faire simple, on entend par paquet, un fichier compressé ou archive, contenant un logiciel ou une librairie, sa documentation associée, et ses dépendances[2]. Un système de paquets binaires est donc un programme qui permet d'installer facilement des logiciels et leurs dépendances associées. Tous ces paquets peuvent être installés à partir de cd-rom ou des dépôts ftp.

Ce qui nous intéresse est bien sûr le concept de distributions audio, dites multimédia. Il est possible de faire de la M.A.O. avec n'importe quelle distribution à partir du moment où on retrouve certains composants essentiels :

- un noyau (ou kernel) optimisé pour le temps réel afin de garantir de faibles temps de latence.
- les drivers Alsa ou freebob (pour le Firewire)
- le serveur JACK
- une collection complète d'applications M.A.O.
- de la documentation

Bien qu'il ne soit pas nécessaire d'avoir une distribution dédiée pour faire de la M.A.O. sous Linux, car tout est configurable sur mesure, cela peut s'avérer intéressant lorsque l'on débute. Les distributions comme 64studio, Musix, UbuntuStudio, Fedora (avec les paquets Planet CCRMA), proposent des configurations spéciales multimédia. Cela comprend un noyau temps réel et tous les logiciels dont nous avons besoin : M.A.O., graphisme, vidéo, bureautique, Internet, etc.

1 Un paquet (ou package en anglais) est un fichier compressé contenant tous les fichiers nécessaires à l'installation d'un programme. Un paquet peut être un logiciel complet, ou une simple librairie. Il existe différents formats de paquet : rpm inventé par Red Hat et utilisé par ses dérivées ; deb utilisé par Debian et ses dérivées ; ebuild utilisé par Gentoo.

2 Une dépendance exprime des relations entre paquets. Pour installer un logiciel on aura besoin de librairies utilisée par ce logiciel. Les dépendances permettent d'installer toutes les dépendances d'un logiciel afin que celui-ci s'installe et fonctionne correctement.

Noyau Temps Réel :

Le noyau Linux est à la base relativement performant (système multi-tâches, gestion de la mémoire, etc.). Il est néanmoins possible d'améliorer encore ses performances, afin d'obtenir en particulier une latence très basse, appréciée en M.A.O. (de l'ordre de la milliseconde si la configuration matérielle le permet). Ceci peut être intéressant pour l'enregistrement ou le jeu en temps réel.

Pour cela il faut appliquer un patch au noyau. Ce patch est développé par Ingo Molnar et permet d'accéder à des options spécifiques. Les distributions dédiées multimédia installent par défaut un noyau optimisé. Pour les autres, il faudra le construire soi-même.

Drivers Audio :

Sous Linux les drivers des cartes sons audio-numériques sont développés par deux projets communautaires.

The Advanced Linux Sound Architecture (ALSA) fournit des fonctionnalités audio et MIDI au système d'exploitation Linux et supporte de nombreuses interfaces audio, des cartes son grand public aux cartes audio-numériques professionnelles. Malheureusement, la compatibilité dépend souvent de la bonne volonté des constructeurs, qui parfois ne désirent pas fournir les informations utiles au développement des pilotes sous Linux. ALSA est développé sous les licences GPL et LGPL et en arrivant sur le site on peut lire le message suivant :

> *« We need users to use, test and provide feedback, programmers to work on low level drivers, writers to extend and improve our documentation, and application developers who choose to use ALSA as the basis for their programs. If you are interested, please subscribe to a mailing list.»*[a]

Chacun peut donc participer à l'élaboration des drivers ALSA, qui permettent d'utiliser la plupart des cartes son intégrées, PCI et même USB. De plus ALSA met à disposition plusieurs outils et utilitaires et notamment des mixers spécifiques à certaines cartes son.

Un autre projet initié par Pieter Palmers, a vu le jour début 2005. Le projet Freebob qui sera prochainement remplacé par FFADO (Free Firewire Audio Drivers), permet de faire fonctionner les cartes son Firewire sous Linux. Moins souple qu'ALSA, car plus récent et pas tout a fait abouti, il permet tout de même de faire de la M.A.O. sous Linux. A ce jour, quelques cartes seulement sont compatibles, mais le projet devrait prendre de l'importance et à terme être intégré aux drivers ALSA.

Le serveur JACK :

Un des points forts de Linux, c'est l'existence d'un logiciel appelé JACK, acronyme de Jack Audio Connection Kit. C'est un serveur audio à faible latence qui permet à plusieurs applications d'utiliser le même périphérique audio et de partager les ressources audio. JACK a été créé pour des applications audio professionnelles et sa

a Citation de la page d'accueil du site d'ALSA : http://www.alsa-project.org/main/index.php/Main_Page

conception se concentre sur deux points clés : exécution synchrone de tous les clients (applications connectées à JACK), et toutes les opérations se font à faible latence.

La puissance de JACK réside dans le fait que l'on peut connecter et inter-connecter toutes les entrées/sorties audio et MIDI des logiciels, comme on le souhaite. Par exemple, on a une boite à rythme et un lecteur de samples[1] qui sont connectés à un effet de Delay[2]. Puis la sortie de l'effet est renvoyée vers un logiciel d'enregistrement. Le tout peut être redirigé vers un logiciel permettant de diffuser en temps réel, ce que vous jouez, sur votre site Internet. Cette grande souplesse permet de créer des configurations complexes avec ses applications préférées : un véritable studio virtuel.

Serveur LASH :

Comme on l'a vu, le serveur JACK très utile, est le logiciel au centre de la M.A.O. sur Linux. Mais il devient très fastidieux de rouvrir une session quelques semaines plus tard au vu du nombre de logiciels pouvant être utilisés, chacun ayant ses propres fichiers de sauvegarde. C'est pour cela que le serveur LASH a été développé. Ce logiciel permet de gérer une session faite de plusieurs applications « jackifiées » et ainsi de pouvoir sauver/restaurer facilement l'état de tout l'ensemble du studio. Malheureusement, cela nécessite que les logiciels supportent cette application et ce n'est le cas que pour un petit nombre d'entre eux. En attendant, il faut donc avoir une mémoire à toute épreuve, ou une très bonne organisation dans son travail et dans ses sauvegardes.

Les plugins LADSPA :

Sous Linux, il existe aussi de nombreux effets ressemblant aux VST/VSTI sous Windows. Les VSTI (Virtual Studio Technology Instrument) sont des instruments de musique ou des effets virtuels au standard VST, définis par Steinberg notamment pour son célèbre logiciel Cubase. Bien que ces instruments virtuels puissent être utilisés sous Linux, la licence n'est pas libre : la distribution est interdite. Aujourd'hui de nombreux logiciels sous Linux fonctionnent avec les VST. Mais pour les utiliser, on est obligé de les activer soit même.

Les LADSPA (Linux Audio Developer's Simple Plugin API) sont aussi des effets virtuels mais supportés nativement sous Linux. Il en existe des centaines et la plupart des logiciels les intègrent parfaitement : reverbs, delay, filtres, compresseur, gate, limiteur, expandeur, etc.

Certains LADSPA sont des générateurs de son (sinus, noise, signal triangle, carré) et sont utilisables dans des logiciels de synthèse modulaire, tels que Alsa Modular Synth, Ingen, Galan, Spiral Synth Modular. Un synthétiseur modulaire permet de connecter librement des modules entre eux : générateur de son, effets, générateur d'enveloppe, VCO, VCF, VCA (Voltage Controlled Oscillator, Filter, Amplitude) et ainsi de créer ses propres instruments virtuels, ce qui autorise une grande créativité.

---

1 Un échantillon (sample en anglais) est un extrait de musique ou un son réutilisé en dehors de son contexte d'origine afin de recréer une nouvelle composition musicale. L'extrait peut être une note ou un motif musical.

2 Une chambre d'écho (ou delay en anglais) est un dispositif généralement électronique utilisé en sonorisation pour retarder un son donnant une impression d'écho. Le son se répète, selon le calibrage, indéfiniment ou s'épuise au bout d'un certain temps.

Vers une nouvelle norme de plugins :

Les plugins LADSPA devraient à terme être remplacés par les plugins LV2. Les plugins LV2 sont de petits bouts de programmes permettant de traiter ou de générer un signal audio. Ce ne sont pas des logiciels à part entière. En effet, un plugin ne peut pas fonctionner tout seul, il faut le charger dans un programme appelé "Hôte" (ou host en anglais). Tout comme les plugins LADSPA, ils peuvent être utilisés :

- en tant qu'effets temps réel dans les logiciels d'enregistrement audio ou de composition de musique "live";
- en générateur de sons dans des synthétiseurs modulaires ;
- en processeur d'effets différés, dans les logiciels d'édition du son.

Plugins DSSI :

Les plugins DSSI, sont l'équivalent des VSTI : des instruments virtuels. Ce sont souvent des synthétiseurs mais rien n'empêche de concevoir un DSSI pour les traitements sonores. Il existe par exemple un DSSI oscilloscope. Ces plugins sont supportés par certains logiciels, mais peuvent être utilisés tout seul en mode « standalone ». Cela signifie qu'ils n'ont pas besoin d'être utilisés dans un logiciel hôte pour fonctionner.

Protocole OSC[a] :

L'OSC ou Open Sound Control permet la transmission d'informations à travers un réseau en utilisant le protocole UDP[1]. La communication peut se faire entre des logiciels installés sur le même ordinateur ou bien sur deux machines différentes. Ce protocole est très rapide et ne provoque pas de latence perceptible lors du passage entre deux logiciels. Il remplacera certainement le MIDI non seulement parce qu'il offre les mêmes possibilités mais aussi parce qu'il est beaucoup plus puissant. Les données transmises peuvent servir à la synchronisation, au transport, mais aussi au contrôle des paramètres, à la création de nouveau fichier, sauvegarde, etc. Ainsi dans un logiciel tel que PureData, il est possible, à travers le protocole OSC, de créer de nouveaux patchs, de connecter des modules entre eux.

b) Les logiciels

En général, les personnes qui veulent passer à Linux et qui avaient l'habitude de travailler sur un système d'exploitation propriétaire, se demandent quels logiciels équivalents ils pourront utiliser et s'ils rempliront les mêmes fonctions. Il est vrai qu'on retrouve un certain nombre de solutions similaires. Certains sont encore en grande phase de développement, d'autres sont déjà bien avancés et peuvent rivaliser sans crainte contre leurs homologues propriétaires. Nous verrons donc quelques logiciels phares et déjà bien aboutis sous Linux qui permettent une production/composition musicale de qualité.

Logiciels généralistes :

a Site officiel : http://opensoundcontrol.org/about

1 Le User Datagram Protocol (UDP, en français protocole de datagramme utilisateur) est un des principaux protocoles de télécommunication utilisé par Internet.

- Rosegarden :

Rosegarden est un séquenceur MIDI, audio et un éditeur de partition. Il permet d'enregistrer, d'arranger et de composer de la musique, sous forme de partitions traditionnelles, avec des données MIDI, ou des fichiers audio importés ou enregistrés à partir d'un microphone ou de n'importe quelle source audio. On peut utiliser Rosegarden pour écrire, éditer et organiser sa musique afin de créer une composition sur laquelle on peut ajouter des effets, la mixer puis l'exporter en gravant un CD ou dans un format de fichier permettant la diffusion sur Internet. Rosegarden contient aussi un système de notation professionnelle pour la création et l'édition de partition permettant de réaliser des impressions de bonne qualité.

A l'origine, Rosegarden était un projet universitaire mené par deux étudiants à l'Université de Bath à la fin de l'année 1993 : Chris Cannam travaillait sur l'édition de partition et Andy Green sur le séquenceur MIDI. Depuis 1995, d'autres développeurs se sont attachés au projet et la première version officielle (0.1) est sortie en Juin 2001. La version actuelle est la 1.5.1 (Mars 2007) et montre de nombreuses évolutions, au point de vue de l'ergonomie, des fonctionnalités.

- Muse :

Muse existe depuis 2003 et est l'autre grand séquenceur midi avec Rosegarden. C'est donc un séquenceur MIDI et audio avec des fonctions d'enregistrement et d'édition. A la différence de Rosegarden, il rend possible l'édition audio mais par contre, il ne peut pas imprimer de partitions. Il dispose d'un vrai mixeur audio avec bus et possibilités d'effets. Ce logiciel vise à être un studio virtuel multipiste complet pour Linux.

Ces deux logiciels sont de véritables alternatives aux solutions propriétaires tels que Cubase ou Logic audio. Même s'il leur reste un peu de chemin avant d'être aussi performant, ils sont déjà à un stade bien avancé et permettent de réaliser les fonctions essentielles : enregistrement, arrangement, mixage, application d'effet LADPSA, intégration des VST, et des DSSI pour Rosegarden.

- LMMS :

LMMS à pour but d'être une alternative gratuite aux logiciels tels que FruityLoops, Reason, permettant de produire de la musique électronique en créant des boucles à l'aide de samples ou de synthétiseurs virtuels. Même si ce logiciel est en grande phase de développement, il est utilisable et permet d'ores et déjà d'utiliser les effets LADSPA, et d'importer des plugins d'instruments VST. Il intègre un séquenceur sous forme de Pattern / Song. Un Pattern est « un motif », un bout de séquence. Un Song, est un assemblage de Patterns pour former un morceau complet. Par exemple, on peut créer un pattern pour la rythmique, un pour la basse et un autre pour la mélodie ainsi que des variations pour chacun. Puis on les importe dans le Song et on peut les séquencer comme on le désire : superposition ou enchaînement de plusieurs Patterns. On peut ainsi créer des morceaux complexes.

- Hydrogen :

Hydrogen est une boite à rythme. Ce logiciel est donc destiné à créer des boucles rythmiques à l'aide d'un séquenceur sous forme de Pattern / Song comme dans LMMS. On peut créer des kits de batteries complexes avec ses propres sons grâce à un système de couches : on peut utiliser plusieurs sons en fonction de la vélocité pour plus de réalité. En effet lorsque qu'un vrai batteur joue et qu'il tape plus ou moins fort sur son

instrument, le son changera en fonction de la force du coup : le volume sonore et le timbre de l'instrument. En MIDI, la vélocité correspond au volume sonore (de 0 à 127). Dans Hydrogen on peut découper chaque son (grosse caisse, caisse claire, etc.) en plusieurs couches : de la vélocité 0 à 60, je joue tel son de grosse caisse ; de la vélocité 61 à 80, je joue un son de grosse caisse qui contient plus d'attaque ; de 80 à 100, je joue une grosse caisse avec encore plus d'attaque etc. J'ai donc la possibilité de créer des kits de sons avec un comportement proche d'une véritable batterie.

Enregistreurs multipistes / Stations de travail Audio-Professionnelles :

- Ecasound :

Ecasound est un logiciel audio-numérique développé depuis 1995 par le tchécoslovaque Kai Vehmanen. Il s'agit donc d'un logiciel très mature et très complet, qui peut être utilisé aussi bien pour faire de l'enregistrement multipiste que du mixage, de l'arrangement ou du traitement audio. Même s'il ne possède pas d'interface graphique native, il peut être contrôlé de manière interactive au moyen d'instructions envoyées en ligne de commande ou par l'intermédiaire de scripts (petits programmes). Cette façon de procéder peut paraître très contraignante et peu intuitive, mais elle a ses avantages car cela permet d'envoyer une multitude d'instructions simultanément, ce qu'une souris est incapable de faire. Il existe tout de même des projets parallèles qui ont développé pour Ecasound une interface graphique comme par exemple « Visecas ». D'autre part, Ecasound est totalement contrôlable via MIDI, ce qui en fait un très bon logiciel pour les performances en temps réel car c'est l'interface graphique qui généralement demande le plus de ressources.

- Ardour :

Il existe de nombreux autres projets dont certains proposent déjà des fonctions d'enregistrement multipistes et de lecture simultanés mais il est encore trop tôt pour parler de véritables stations de travail audio-professionnelles. On pourrait tout de même citer : Traverso, Protux, XO Wave, Qtractor.

De ce point de vue, Ardour est la référence absolue en matière de production audio-professionnelle sous Linux, d'autant que la version 2.4 du logiciel est sortie au mois d'Avril 2008. Les fonctionnalités d'Ardour sont nombreuses : enregistrements multipistes, édition non-destructive (lorsque l'on modifie un son, l'original est toujours présent), une table de mixage, tout est automatisable, création de pistes, bus et plugins illimités, synchronisation au Timecode, et reconnaissance des surfaces de contrôles telles que la Mackie Control Universal ou la Behringer BCF 2000. La version 3 du logiciel devrait intégrer le MIDI (enregistrement, édition, etc.).

Ce projet débuté en 2000 est développé par des personnes du monde entier aidées par un réseau de musiciens et d'ingénieurs du son prêts à tester, rapporter les bugs et proposer de nouvelles idées. On peut dire qu'Ardour s'efforce de répondre aux besoins des utilisateurs professionnels. Ce logiciel est sous licence GNU GPL, donnant ainsi aux utilisateurs la liberté de le modifier, et de le redistribuer.

Langage de programmation audio :

En ce qui concerne le langage de programmation audio sous Linux, il existe deux solutions : le langage graphique et le langage écrit. Ces logiciels disposent d'un langage spécifique, dédié au développement d'applications audio. Certains offrent de très puissants outils de programmation et permettent de créer des synthétiseurs, séquenceurs, effets. Sans entrer dans les détails on peut citer les plus utilisés d'entre eux :

Pure Data (similaire à Max MSP) est un logiciel de programmation en mode graphique permettant la réalisation de synthétiseurs, d'effets visuels et de toutes autres réalisations liées au son, au graphisme et à la vidéo. Sa conception modulaire permet la création de projets d'une façon très rapide par l'assemblage de boites reliées entre elles et formant un réseau. Il permet de plus un traitement en temps réel et interactif entre différentes sources (clavier, capteur, surface de contrôle...). Dans la même lignée on pourra citer Jmax et OpenMusic développés par l'IRCAM.

Des logiciels comme Csound ou SuperCollider, sont des langages écrits de programmation fournissant de nombreuses fonctions dédiées au traitement du signal et à la synthèse sonore. Ils offrent beaucoup plus de possibilités qu'un logiciel de programmation graphique, mais demande aussi plus de connaissances en programmation et donc plus de temps d'apprentissage.

Synthétiseurs Modulaire :

Il existe aussi un certain nombre de synthétiseurs modulaires. Comme expliqué plus haut, dans ce type de logiciel, on a à disposition des modules, chacun remplissant des fonctions différentes : générateurs de sons, d'enveloppes, conversion midi/fréquence, effets, etc. Il est possible de connecter les modules entre eux afin de créer des synthétiseurs originaux.

Alsa Modular Synth en est un bel exemple. Disposant de modules natifs, il intègre aussi les plugins LADSPA ce qui lui donne des possibilités quasiment infinies. Avec ses modules d'entrée et de sortie, il est tout a fait possible de créer un effet, qui pourra être connecté à d'autres logiciels via le serveur JACK. De plus Alsa Modular Synth est entièrement contrôlable en MIDI.

Un autre synthétiseur modulaire actuellement encore en développement mais très prometteur : Ingen. Il intègrera en plus les plugins LV2 et sera entièrement géré avec le protocole OSC. Ce qui permettra de le contrôler à distance, de créer des patchs, de connecter les modules entre eux à l'aide de n'importe quel autre logiciel et ceci à travers un câble réseau éthernet.

Le Mastering :

Même si sous Linux, il manque encore des outils vraiment performants pour le mastering, certains malgré tout permettent d'en faire. Quelques plugins LADSPA sont utilisables sous Ardour : limiteurs, compresseurs, égaliseurs. Rezound, peut être utilisé pour le mastering. Il offre de nombreux outils pour finaliser un fichier son : compression dynamique, normalisation, noise gate. Il reconnaît aussi les plugins LADSPA.

Mais le logiciel le plus utilisé est sans doute Jamin. En effet, c'est un logiciel complet offrant un égaliseur 3 bandes, un compresseur 3 bandes, un limiteur et un visualisateur de spectre. Il est compatible avec le serveur JACK. On peut donc le connecter aux sorties d' Ardour et enregistrer le résultat sur une piste de ce dernier.

IV. Création d'une Formation M.A.O.

J'ai maintenant toutes les clés en main pour envisager la création d'une formation. Le choix des logiciels libres est un choix personnel qui me semble pertinent : les autres formations en M.A.O. utilisent exclusivement des logiciels propriétaires. Ce choix s'inscrit aussi dans une philosophie et une démarche de transmission des connaissances et d'un savoir faire. Tout ce qui est nécessaire à la création musicale sous Linux existe aujourd'hui et permet d'envisager la création d'une formation complète : montage, mixage, mastering, audio, MIDI. Il reste bien sur de nombreux progrès à faire mais les logiciels libres prendront sans aucun doute de plus en plus d'importance dans les mois et les années qui viennent. Je parlerai dans ce chapitre de la manière dont je pense créer une formation en répondant à certaines questions fondamentales sur la pédagogie, à qui est elle sera destinée, quel en sera le contenu. Le programme complet sera établi et le premier module détaillé sera disponible aux formats PDF et texte, sous licence Creative Common dans un document annexe.

IV.1 - But de la formation : Pourquoi, pour qui, comment ?

a) Pourquoi ?

Cette idée part de deux constats. D'une part les logiciels libres prennent petit à petit de l'importance, prônant un modèle de libre diffusion de l'information et de l'immatériel qui fédèrent de plus en plus de personnes, scientifiques, chercheurs, professionnels, amateur. D'autre part, ces cinq dernières années ont vu fleurir un nombre important de formations M.A.O. sur des logiciels propriétaires tels que Cubase, Pro-Tools, Reason ou Logic répondant à une forte demande de la part d'intermittents, désireux d'élargir leurs compétences ou de musiciens amateurs ayant besoin de connaissances particulières pour enregistrer leur groupe de musique, ou monter leur propre « Home Studio ».

Lorsque l'on voit l'évolution et l'engouement pour les systèmes d'exploitations Linux et de ses logiciels, on peut penser qu'il serait intéressant de réfléchir dès aujourd'hui à la question d'une formation. D'ores et déjà, l'école de son, SAE, mondialement connue, s'est intéressée au projet Ardour qui est un DAW (Digital Audio Workstation) au même titre que Pro-Tools. Depuis Mai 2007 cette école soutient et sponsorise ce projet. Une autre école en Serbie s'est équipée de 4 stations de travail entièrement sous Linux avec la distribution Planet CCRMA.

Cette formation doit répondre à un nouveau besoin. Elle nécessite de prendre en compte plusieurs paramètres, et de se poser au préalable un certain nombre de questions :

- A qui sera destinée la formation ?
- Dans quel ordre aborder les différentes parties ?
- Qu'est ce qu'on étudie en fonction du niveau des élèves ?

Ces questions sont importantes pour moi en tant que formateur afin d'adopter la meilleure pédagogie et pour les élèves pour que la formation réponde exactement à leurs

besoins.

b) Pour qui ?

J'aimerai proposer une formation destinée au plus grand nombre :

- Personnes désirant se mettre à l'informatique musicale, pour composer/enregistrer leur musique.
- Amateurs ou professionnels ayant déjà de bonnes connaissances en M.A.O. sous Windows ou Mac et souhaitant passer à Linux.
- Groupes de musique, studios associatifs, studios professionnels, voulant enregistrer des groupes et produire des albums tout en utilisant des logiciels Open Source et des licences libres.
- Aux industries et organismes qui désirent migrer vers Linux une partie ou la totalité de leurs systèmes audio.
- Aux artistes indépendant et PME qui font manipuler leurs fichiers audio en sous-traitance et qui aimeraient gagner en autonomie.

Selon le profil des élèves, il sera judicieux de proposer des formations différentes, adaptées à leurs connaissances. En effet, on ne peut pas proposer la même formation à quelqu'un qui a déjà de bonnes bases et à quelqu'un qui n'y connaît rien du tout. Voici plusieurs degrés de prise en compte du niveau :

- Connaissances de l'informatique en général
- Connaissances de Linux
- Connaissances en Musique Assistée par Ordinateur
- Connaissances en techniques du son et traitement du signal (Enregistrement, mixage, montage, effets, etc.)
- Connaissances en musique ou en travail sonore.

Un élève confirmé, devra avoir certaines connaissances sur lesquelles il ne sera pas nécessaire de revenir. L'étude de fonctionnalités plus poussées sera alors envisagée. Un débutant sera obligé de passer par certaines étapes avant d'entamer l'apprentissage de la M.A.O.

On peut alors imaginer de découper la formation en différents modules, qu'on traitera en fonction du niveau et du besoin des élèves. On peut aussi imaginer de proposer un système de programme à la carte, où l'élève choisit parmi les modules, ceux qu'il souhaite aborder.

Lors d'une inscription il sera donc demandé le niveau de chaque élève selon les critères précédemment cités, mais aussi leur configuration matérielle : PC, Mac, cartes son, cartes vidéos, etc. de manière à prévoir les éventuels problèmes que l'on pourrait rencontrer au début (installation, reconnaissance de matériel,etc.)

En plus de la formation, il me semble important d'apporter d'autres services, comme le conseil pour le choix et l'achat de matériels, de l'installation/mise à jour et de la maintenance de systèmes Linux dédiés à l'Audio.

c) Le programme

Voici maintenant le plan et le résumé du contenu des différents modules qui pourront être abordés :

Module 1 : Installation et configuration du système.
Module 2 : Manipulation de fichiers audio
Module 3 : Prise en main d'Ardour
Module 4 : Enregistrement multipiste
Module 5 : Introduction au MIDI
Module 6 : Production de musique électronique
Module 7 : Le mixage
Module 8 : Le mastering

Module 1 : Installation et configuration du système.

1. Choix de la distribution
2. Installation du système d'exploitation Linux. Dual boot[1] avec Windows ou Mac
3. Maintenance du système : Organisation de l'arborescence et des fichiers, le terminal, configuration système.
4. Paramétrage et installation des logiciels/paquets nécessaires, et du noyau temps réel.
5. Les drivers ALSA/Freebob : présentation, configuration.
6. Présentation et configuration du serveur audio JACK

Module 2 : Manipulation de fichiers audio

1. Explication des différents formats de fichiers (mp3, Wav, ogg, wav64, Aiff, flac, etc.)
2. Installation des codecs : lecture de mp3, ogg, etc.
3. Extraction de fichier audio provenant d'un CD
4. Conversion, ré-échantillonnage : Audacity / Rezound.

Module 3 : Prise en main d'Ardour

1. Création d'une session.
2. Présentation de l'environnement du logiciel, des menus, et options.
3. Importation de son.
4. Montage multipiste avec « layering » et « crossfades ».

Module 4 : Enregistrement multipiste avec Ardour

1. Présentation de la fenêtre de mixage.
2. Création de pistes, de bus et routage des différentes sources et sorties avec JACK.
3. Enregistrement d'une source interne (synthétiseurs, ou autre logiciel) et à partir d'un micro.
4. Les punch-ins avec les musiciens.

1 Le dual boot ou multiboot est le terme pour désigner la possibilité de démarrer plusieurs systèmes d'exploitation sur un même ordinateur.

Module 5 : Introduction au MIDI

1. Qu'est ce que le MIDI ? A quoi ça sert ?
2. Utilisation de base du MIDI.
3. Synthétiseurs matériels et logiciels.
4. La gestion des périphériques MIDI avec Jack.
5. Utilisation des « Soundfonts » avec Qsynth.
6. Le mixage audio et MIDI avec Rosegarden et Ardour.

Module 6 : Production de musique électronique

1. Fabrication de sons et d'effets sonores (ZynAddSubFX, PHASEX).
2. Création de rythmes avec Hydrogen.
3. Jeu en temps réel : Utilisation d'un contrôleur MIDI avec TerminatorX, Seq24, Specimen.
4. Introduction à LMMS.

Module 7 : Le mixage

1. Rappel sur les notions de compresseurs, limiteurs, gate, expandeur, équalisation, réverbs, etc.
2. Utilisation des effets LADSPA.
3. Utilisation des automations et de ses différents modes (manuel, play, write, touch).

Module 8 : Le mastering

1. Présentation du logiciel de mastering : Jamin (Jack Audio Mastering INterface).
2. Routage du signal entre Ardour, Jack et Jamin.
3. Principes de base pour augmenter la "loudness" d'un projet.
4. L'utilisation des scènes avec Jamin.
5. L'enregistrement du signal masteurisé.

Par la suite j'aimerai mettre en place un module plus spécialisé dans la programmation musicale. Avec des logiciels orientés langage graphique comme Pure Data, ou encore plus poussé tels que SuperCollider, Csound ou Chuck. Ces logiciels demandent un effort plus important d'apprentissage car ce sont de véritables langages de programmation, mais ils permettent aussi d'aller beaucoup plus loin dans la création de musique électronique. On peut notamment créer ses propres synthétiseurs, lecteurs d'échantillons, ses effets et combiner le tout dans un seul programme. Pure Data permet entre autre, l'utilisation de capteurs et tout type de contrôleur externe, de gérer en MIDI ou en OSC et aussi d'être couplé à d'autre langages de programmation tels que Python, PERL ou JAVA.

d) La pédagogie

Le rôle du formateur est de transmettre ses connaissances de la manière la plus simple et la plus compréhensible à ses élèves. Ceci se révèle être un exercice difficile. Voici une liste de quelques points qui pourraient m'aider à mener correctement un cours :

- Adapter son vocabulaire pour que les explications ne paraissent pas trop simples ou trop compliquées et ceci afin que chacun s'y retrouve : éviter l'élève qui s'ennuie parce que c'est trop basique ou l'autre qui nage complètement parce que c'est trop technique.
- Préparer les cours en fonction du matériel utilisé par les élèves. Mac /PC, carte vidéos, carte son, etc. Ceci permettra de déceler, s'il y a lieu, les éventuels problèmes que l'on pourrait rencontrer lors de la formation et ainsi ne pas perdre de temps à chercher des solutions pour se concentrer exclusivement sur l'apprentissage de la M.A.O.
- Vérifier la compréhension du cours par les élèves par de petits exercices pratiques, des QCM en fin de chapitres portant sur les notions principales à appréhender. Ceci présente l'avantage de se rendre compte immédiatement de la bonne compréhension du cours et de s'assurer que les notions élémentaires sont acquises. Le bon déroulement de la suite du cours n'en sera qu'amélioré.
- Un bilan à la fin de chaque module sera effectué. On reprendra rapidement les différents points abordés dans le cours et les éventuelles questions seront traitées. Ceci permet d'avoir une vision globale de chaque module et de mieux cerner la cohérence et la logique des différents chapitres.

La pédagogie adoptée devra aussi être différente si l'on s'adresse à des enfants ou des adultes. En ce qui concerne la formation que je veux mettre en place, elle sera principalement destinée à des adultes. Dans ce cas, il faut noter qu'un adulte a besoin de :

- savoir où il va pour assimiler : le plan et les objectifs de la formation doivent être clairement définis. Il faut mettre en avant la logique suivie entre les différentes parties de la formation ;
- s'appuyer sur son expérience pour se retrouver : la connaissance doit sembler surgir des connaissances passées, être une adaptation de ce qui est déjà connu ; l'enseignement doit être interactif.

L'adulte s'engage dans une formation avec une attente précise, la place du formateur est primordiale. La formation comporte plusieurs parties, pas toujours évidentes à cerner comme étant utiles pour son projet. Il est donc nécessaire d'expliquer et de convaincre sur la logique suivie et sur l'ordre des chapitres abordés. Il me semble aussi qu'une bonne approche pour l'apprenant est de s'appuyer sur des techniques de participation active que l'on peut mettre en place sous forme de bilans personnels, d'échanges interactifs ("qu'en pensez-vous ?"), de mises en situations pratiques. Cette technique sera bénéfique pour l'élève car il se sentira impliqué et par conséquent motivé.

Étant moi même autodidacte, je trouve important de donner les clés essentielles à l'apprenant pour qu'il puisse continuer seul. Ceci passe par des techniques

d'apprentissages spécifiques mais aussi par l'utilisation des ressources disponibles telles que les forums, les sites Internet et les livres. Il faut distinguer les savoirs instruits, des savoirs construits par l'élève. En effet les savoirs instruits sont ceux apportés par le formateur, tandis que les savoirs construits font appel à l'autonomie de l'apprenant et seront assimilés plus rapidement.

En fin de formation je proposerai un petit questionnaire de satisfaction en vue d'améliorer le contenu, la manière dont sont traités les différents modules, etc. Une façon de parfaire la formation pour les élèves suivants.

e) Mise en place d'une formation

Plusieurs possibilités s'offrent à moi pour mettre en place une formation :

- Création d'une association ou une micro entreprise
- Passer par une structure existante déjà spécialisée
- Démarchage auprès des écoles de musique

Pour débuter il serait peut être judicieux de passer par une structure déjà existante spécialisée en formation professionnelle. Je pourrai ainsi leur proposer d'élargir leur activité avec la mise en place d'une formation sur les logiciels libres. Cela me permettrait de réduire toutes les démarches administratives et de me concentrer sur le contenu et la pédagogie des cours. Une autre possibilité serait de démarcher auprès des écoles de musique, de plus en plus ouvertes à l'outil informatique dans la création musicale.

Dans le cas de la création d'une association ou d'une entreprise il est possible de passer par des organismes certifiés par l'État ou d'obtenir une convention de la région comme par exemple au CFPMS de Marseille (Centre de Formation Professionnelle des Métiers du Son). Ceci apporte une bonne crédibilité et une meilleure reconnaissance auprès des professionnels qui désirent s'inscrire à une formation. Le plus connu est sans doute l'AFDAS (Assurance Formation Des Activités du Spectacle).

L'AFDAS est agrée par l'État pour collecter les contributions obligatoires des entreprises au titre de la formation et répondre aux demandes de financement des entreprises, salariés et intermittents du spectacle. En matière de formation professionnelle, l'AFDAS dispose d'une exclusivité nationale pour la gestion des contributions formations des entreprises relevant du spectacle vivant, du cinéma, de l'audiovisuel, de la publicité, des loisirs et de la distribution directe.

Il y a plusieurs intérêts pour un organisme d'obtenir un conventionnement AFDAS. Le coût d'une formation étant élevé, cela permet à des personnes d'y accéder à faible coup voir gratuitement sous certaines conditions. Le centre de formation a ainsi plus de chances de recruter des élèves. En effet, un particulier peut difficilement s'inscrire à une formation de 2 semaines s'il ne bénéficie d'aucune aide.

De plus, obtenir un conventionnement signifie une certaine qualité dans la formation proposée et une certaine reconnaissance des professionnels. L'AFDAS a en effet des critères stricts de sélection.

Pour obtenir une certification, il faut dans un premier temps justifier du statut de formateur grâce au numéro de déclaration d'activité. Ce numéro s'obtient auprès de la Direction Départementale du Travail, de l'Emploi et de la Formation Professionnelle

(DRTEFP). Il permet de retirer un dossier auprès de l'AFDAS pour leur proposer une formation. Ce dossier devra détailler la formation de la manière la plus complète :

- un programme détaillé de la formation
- le lieu, la date, le nombre d'heures
- le nombre d'élèves
- les modalités d'évaluation et de suivi
- le budget prévisionnel qui inclut : le salaire, les frais de fonctionnement (location ou amortissement du matériel), les fournitures, frais divers.

Enfin, le dossier passera par une commission qui jugera de la pertinence de la formation, de sa faisabilité et acceptera ou non de subventionner le projet. A la fin du stage, on envoi une facture à l'AFDAS, qui rembourse les différents frais afférents à la formation.

En travaillant dans une école de musique, je n'aurai pas à me soucier de ce côté administratif. En effet, pour passer en commission AFDAS, il y a des dates à respecter, car il n'y en a pas tout au long de l'année. Il faut donc préparer les dossiers à l'avance et déterminer les dates du stage et le nombre d'élèves, ce qui n'est pas toujours prévisible.

IV.2 - Module 1 : Installation et configuration du système.

Le détail de ce module est présenté dans un document joint. Il reprend sous forme d'un tutoriel toutes les étapes à suivre pour installer une distribution Linux et la configurer pour une utilisation Audio. Ici je m'efforcerai d'expliquer l'organisation des différentes parties de ce module et mes choix en terme de contenu.

a) Choix de la distribution

Sous Linux il existe de nombreuses distributions permettant de faire de la musique. Cependant mon choix se porte sur une distribution Fedora avec les paquets Planet CCRMA. Après avoir testé de nombreuses distributions, je me suis tourné vers cette dernière, peu connue mais néanmoins excellente.

Fedora, projet sponsorisé par RedHat fournit tous les paquets et logiciels généralistes de base. On pourra donc faire de la bureautique, de l'Internet, du jeu. Planet CCRMA est un projet universitaire Américain visant à fournir tous les logiciels nécessaires pour créer de la musique et travailler l'audio sous Fedora.

La force de Planet CCRMA réside dans la mise à jour très régulière des paquets qui sont disponibles rapidement pour les architectures 32 et 64 bits[1]. Les

1 **L'architecture** correspond à la taille des adresses manipulées par un microprocesseur. Un microprocesseur à architecture 32 calculera ses adresses sur 32 bits. L'avantage d'une architecture 64 bits par rapport au 32 bits est qu'un même processus peut adresser 2^{64} 4 milliards de fois plus.

mainteneurs sont très réactifs. De plus, de nombreux développeurs utilisent cette distribution, facilitant ainsi sa compatibilité (ex : Ardour, Freebob/FFADO, PHASEX, etc.), et elle est connue pour être utilisée par l'IRCAM.

Naturellement, les personnes désirant ou ayant déjà installé une autre distribution ne seront pas exclues. Bien que les systèmes d'exploitations soient légèrement différents, les configurations restent très proches et les logiciels utilisés seront les mêmes. J'inciterai donc les élèves à installer une distribution Fedora/Planet CCRMA mais je ne peux pas les en obliger. Il en est de même pour le choix du gestionnaire de fenêtre : KDE, Gnome, Xfce, fluxbox.

J'ai choisi de mettre sur Internet ce chapitre de mon mémoire, sous la forme d'un site Web reprenant les différentes parties abordées. Ceci me servira de support de cours, de document de référence que je pourrai fournir à mes stagiaires afin qu'ils gardent une trace de ce qu'ils ont appris et afin qu'ils puissent revenir sur telle ou telle partie une fois la formation terminée. Le contenu se trouve à cette adresse dans la section « Guide Audio Linux » : http://mysthr.free.fr/Joomla.

b) Installation du système d'exploitation Linux. Dual boot avec Windows ou Mac

L'installation du système d'exploitation est la première étape. Elle est essentielle et permettra à chaque élève de travailler sur son propre ordinateur. Une séance complète sera donc consacrée à cette étape. Il sera question d'organisation du disque dur, de gestion des partitions[1], de formatage[2] et de configuration de l'installation. Les notions de multi-boot[3] y seront également abordées.

c) Maintenance du système : Organisation de l'arborescence et des fichiers, le terminal, configuration système.

Arborescence :

L'organisation des fichiers est différente sur Windows, Mac ou Linux. Il est important pour les débutants de bien connaître cette organisation pour retrouver facilement les fichiers utilisateurs, les fichiers de configuration des logiciels, les disques durs externes/clés USB et bien comprendre le fonctionnement de son ordinateur. L'arborescence sous Linux répond à une logique bien précise qui sera détaillée dans cette partie.

Notion de Super-Utilisateur :

Ce concept de Super Utilisateur est un peu déroutant au début mais il est essentiel du point de vue de la sécurité du système. En effet, seul le Super Utilisateur

1 En informatique, une partition est une partie d'un disque dur destinée à accueillir un système de fichiers.
2 Le formatage est l'action de formater, c'est-à-dire de préparer un support de données informatique (disquette, disque dur, etc.) en y inscrivant un système de fichiers, de façon à ce qu'il soit reconnu par le système d'exploitation de l'ordinateur.
3 Le **multi-boot** permet de choisir au démarrage de l'ordinateur, quel système d'exploitation on veut utiliser. On peut en installer plusieurs en parallèle. Le logiciel permettant ce choix s'appelle un chargeur d'amorçage ou boot loader.

(administrateur ou root) possède les droits pour modifier, écrire, créer de nouveaux fichiers, etc. Par défaut, un utilisateur n'a aucun droit. Avant de pouvoir configurer le système, installer de nouveaux logiciels, il faut donc comprendre ce qu'est ce Super Utilisateur et comment on peut s'en servir sans courir de risque pour ses données. Il faudra se connecter en tant qu'administrateur à chaque fois que l'on voudra ajouter ou supprimer une application, modifier le système.

La Console :

Linux est basé sur un mode non graphique (shell, console) et un mode graphique (X11) sur lequel viennent se greffer plusieurs interfaces (Gnome, KDE, xfce, ...). Même si l'usage de ces derniers est plus répandu, le mode console de par sa simplicité est encore relativement indispensable. Les commandes s'exécutent rapidement et la plupart des logiciels utilisés sont en fait les interfaces graphiques (front end) de logiciels en mode texte. Il faut donc impérativement connaître quelques commandes de base, telles que : créer un nouveau dossier, lister les fichiers d'un répertoire et naviguer dans l'arborescence. Une bonne maîtrise de la console permet dans certain cas de gagner du temps et des ressources systèmes. Parfois, seule la console permet de résoudre les problèmes, notamment lorsque le serveur X (interface graphique) ne fonctionne plus.

Configuration système :

Les trois notions de bases que je viens de décrire sont indispensables pour la suite. Sous Linux, le choix des logiciels est vaste et chaque utilisateur aura ses préférences. Afin que ce document et cette formation soit « compatible » quelque soit la distribution et les logiciels utilisés, toutes les opérations de configuration, installation et modification du système seront réalisées grâce à la console. Il sera nécessaire de se connecter en tant que Super Utilisateur pour effectuer toutes ces opérations. Dans cette partie j'expliquerai les actions à effectuer une fois l'installation de Fedora terminée : configuration de base, mise à jour. L'installation des composants nécessaires à la pratique de l'Audio sous Linux peut alors commencer.

d) Paramétrage et installation des logiciels/paquets nécessaires, et du noyau temps réel.

Configuration des dépôts :

Avant d'installer les logiciels, il faut indiquer au système où il pourra les trouver. Les dépôts sont des espaces de stockage accessibles via Internet. Il en existe une multitude, certains généralistes, d'autres spécialisés. Il faudra donc configurer correctement ces dépôts et notamment celui de PlantCCRMA qui fournit tous les paquets nécessaires pour faire de la musique sous Linux.

Installation des logiciels :

Ici j'expliquerai dans un premier temps, l'utilisation du gestionnaire de paquet à travers une interface graphique : le logiciel Yumex. Ceci permettra de faciliter l'ajout et la suppression de programmes. Après une introduction théorique les élèves installeront les applications audio et les utilitaires permettant une pratique confortable de la M.A.O.

Le noyau Temps Réel[1] :

Le noyau Temps Réel est un composant très important de l'optimisation de l'ordinateur pour la création musicale sous Linux. J'expliquerai donc brièvement son importance, en quoi il consiste, puis son installation.

e) Les drivers ALSA/Freebob : présentation, configuration.

Les logiciels sont installés, le système fonctionne correctement. Avant de commencer à créer de la musique, il faut configurer la carte son. Il existe deux projets communautaires qui développent des drivers pour Linux. L'un (Freebob) est exclusivement réservé aux cartes sons fonctionnant sur port Firewire. Le deuxième (ALSA) gère tous les autres types de cartes (USB, PCI, etc.). Dans cette partie, je ferai le point sur ces drivers afin d'éviter toute confusion. Puis on abordera l'installation et la configuration, qui demande quelques manipulations supplémentaires pour les cartes son Firewire.

f) Derniers petits détails.

Cette dernière étape de configuration sera utile pour optimiser le système. Ceci permettra de profiter au maximum du Temps Réel et des temps de latence très faibles qu'il fournit. Une fois le système entièrement configuré, il n'y aura plus besoin de revenir dessus et on pourra se consacrer pleinement à la découverte des logiciels et à la création musicale.

g) Présentation et configuration du serveur audio JACK

Avant de partir à la conquête des nombreux logiciels disponibles sous Linux, il faut parler du serveur audio JACK, car il sera le centre névralgique du studio. Toutes les applications, les entrées, sorties audio et midi transiteront par ce serveur. J'aborde ce logiciel à la fin du premier module, parce que sa configuration dépendra beaucoup des choix que l'on aura effectués précédemment. JACK se lance en mode texte, c'est à dire dans une console. Heureusement il existe des interfaces graphiques pour faciliter son utilisation. C'est à travers l'une d'entre elle que je détaillerai chaque option de configuration. Depuis le début de ce module, j'ai tenté d'apporter toutes les clés nécessaires pour que la chaîne ordinateur, système, logiciels, carte son soit la plus optimisée et la mieux adaptée à chaque configuration. Le serveur JACK fait partie de cette chaîne, il est donc indispensable de passer un peu de temps à comprendre son fonctionnement et à se l'approprier pour en tirer le meilleur parti.

1 Source : http://doc.fedora-fr.org/wiki/Fedora_Core_et_le_Temps_R%C3%A9el

IV.3 - Rencontres avec des professionnels

a) Explication de la démarche

Il me semblait intéressant de rencontrer des professionnels de la formation M.A.O. Conscient de la difficulté à trouver des enseignants de M.A.O. sous Linux, j'ai tout de même décidé de contacter des gens afin d'avoir un retour de leur expérience de formateur, sur l'organisation et le déroulement de leur cours. Parmi les personnes rencontrées (voir Annexe 2) j'en retiendrai deux qui ont des approches différentes et qui ont su répondre à mes interrogations. J'ai ainsi pu comparer leur méthode et leur expérience pratique avec la mienne encore purement théorique. Cette comparaison me permettra sûrement d'enrichir quelques points de ma formation et aussi de consolider certaines idées.

b) Didier Merlateau

J'ai dans un premier temps rencontré Didier Merlateau de manière virtuelle sur le site de Linuxmao. C'est un des sites les plus actifs dans le domaine de la musique sous Linux. J'ai appris par hasard qu'il donnait des cours de M.A.O. sous Linux : la rencontre idéale. Je suis allé à Brest pour le rencontrer.

Électronicien de formation, il abandonne ce métier pour ne se consacrer qu'à sa passion: la musique. Utilisateur de Linux depuis maintenant plus de deux ans, il joue régulièrement en concert avec ses synthétiseurs, ses logiciels libres et donne des cours de M.A.O. dans une école de musique. Cela fait maintenant plusieurs années qu'il travaille dans cette école, mais depuis l'année dernière, il dispense des cours uniquement sous Linux.

Son approche est donc différente de celle que j'avais imaginée : il travaille avec de petits groupes d'élèves relativement jeunes, sur une année entière, à raison d'une heure ou deux par semaine et par groupe. De plus, travaillant dans une école de musique, il n'est pas concerné par les conventions que l'on peut obtenir auprès de l'AFDAS ou de la région. Sur le plan matériel, il dispose de plusieurs ordinateurs configurés pour travailler sous Linux, des cartes sons dédiées et autres périphériques (mico, claviers, etc.). Les élèves n'apportent pas leur matériel, tout est organisé sur place. Ses cours sont axés sur la pratique avec une prise en main très rapide des logiciels. Dès le premier cours les élèves s'approprient l'environnement Linux et testent les différents logiciels. Puis après une initiation au serveur JACK, centre névralgique du studio virtuel sous Linux, ils passent rapidement à de petits enregistrements avec Ardour, abordent les notions de MIDI. Il est difficile d'établir un programme très strict car les élèves ont des attentes et des besoins différents en fonction de leur pratique musicale. Certaines notions essentielles sont cependant toujours abordées.

Les élèves les plus motivés installent ensuite un système d'exploitation Linux chez eux. Didier Merlateau apportera son aide dans le choix de la distribution, l'installation, le conseil dans l'achat de matériel, carte son et la maintenance : « *C'est bien de voir que certains s'y mettent vraiment.* » m'a-t-il dit. En effet, au début de cette année un peu spéciale pour lui (c'est la première année qu'il met en place une formation M.A.O. sous Linux), il s'est heurté à des élèves voulant absolument apprendre ou approfondir un

logiciel propriétaire utilisé régulièrement chez eux. Cela peut se comprendre mais "*il existe de très bonnes alternatives en logiciels libres et au moins il n'y a pas besoin de les pirater.*"

Malgré quelques réticences parmi les élèves et parmi certains professeurs assez conservateurs de l'école de musique et souvent plutôt opposés à la musique informatique, Didier Merlateau persévère et fait tout de même des heureux : « *Je ne demande qu'à exister »*. Il envisage de monter prochainement sa propre structure. Il est vrai que Linux n'est pas encore très répandu et les logiciels libres sont souvent associés à « gratuité » qui est malheureusement dans les esprits, synonyme de moins bonne qualité. Mais le nombre croissant d'utilisateurs de Linux incite à l'optimisme. Il organise en outre, un concert au sein de son école de musique à la fin du mois de Mai réunissant des musiciens utilisant des logiciels libres notamment ceux présents sur Linuxmao.org. Ce concert aura pour but de valoriser son travail au sein de l'école et surtout de promouvoir l'utilisation de systèmes d'exploitation libres pour la création musicale.

c) Rolf Entgelmeier de l'AGESCA

L'AGESCA (Actions Globales d'Enseignement et de Soutien Culturel et Artistique) est avant tout un centre de formation musicale (déclaré auprès de la DRTEFP[1]). L'association a proposé pendant de nombreuses années des cours de musique, des stages, des master-classes. Depuis 2001, l'AGESCA a recentré ses actions de formation autour de la M.A.O. Cela fait sept ans maintenant que Rolf Entgelmeier encadre les différents stages proposés par l'association. Il travaille également depuis cette année à l'école Sutdio M où il enseigne aussi la M.A.O.

Contrairement à une école de musique qui fonctionne sur toute une année, l'AGESCA propose ponctuellement des stages sous plusieurs formules :

- Stage d'initiation sur Cubase ou Pro-tools pour apprendre les notions de bases (matériel/branchement, audio/midi, échantillonneur/séquenceur, enregistrement puis composition).
- Stage "Home-studio" uniquement sur Cubase pour un travail plus approfondi sur les outils informatiques de la création musicale. Le travail sur les projets des stagiaires est chose courante.

La durée des stages est souvent de 50 heures réparties sur 2 semaines : « *Je préfère répartir 50 heures sur 2 semaines sinon les journées sont trop chargées et ça fait beaucoup de choses à assimiler pour les élèves »*. Mais la durée peut varier pour des demandes particulières avec des contenus de formations personnalisées comme par exemple leur formule cours particuliers par tranches d'une heure ou le module léger de 10 heures. Les stages sont aussi limités en nombre de places. Comme le dit Rolf Entgelmeier « *Au début, on organisait des stages pour 5 personnes mais maintenant on se limite à 3 voire 4 personnes maximum pour garantir une bonne qualité dans nos formations »*.

L'association dispose de tout le matériel nécessaire pour travailler dans de bonnes conditions : 2 PC, 1 MAC avec cartes sons externes compatibles Pro-Tools, tables de mixages, séquenceurs et échantillonneurs, clavier midi, micro. Les stagiaires peuvent

1 DRTEFP : Direction Régionale du Travail, de l'Emploi et de la Formation Professionnelle.

très bien apporter leur matériel s'ils possèdent les logiciels utilisés lors de la formation. De cette manière ils peuvent travailler avec leurs sons et surtout sur leurs projets créant ainsi une dynamique dans la pédagogie et une corrélation directe entre la théorie et la pratique.

Au delà des stages, l'AGESCA propose un suivi technique et une aide sur le matériel home-studio. Ceci leur permet de garantir un suivi de leurs stagiaires :

- Conseils dans l'achat de matériel
- Installation de logiciels et de différents périphériques
- Optimisation de l'installation
- Aide au diagnostic en cas de panne
- Interventions légères : reconfiguration, défragmentation, récupération des données…

On peut donc dire que leur service de formation est très complet et permet de s'adapter à tous les besoins des stagiaires. Ceci est notamment possible grâce aux conventions passées avec l'AFDAS qui leur permettent d'acheter et entretenir le matériel. Mais le principal but de l'AFDAS est de permettre aux artistes et techniciens du spectacle de faire valoir leurs droits à la formation professionnelle. Ainsi ils peuvent bénéficier d'un nombre d'heure de formation par an et/ou une rémunération ou une allocation calculée sur la base de certains critères tels que le nombre d'heures effectuées durant la dernière année.

Les formations M.A.O. proposées coûtant cher, cela leur permet d'avoir potentiellement plus de stagiaires. Mais « *ces deux dernières années à cause du changement du statut des intermittents, on a de moins en moins de demandes »*. L'année dernière ils n'ont pu proposer que 2 stages de 50 heures.

d) Bilan des rencontres

Ces rencontres ont été très enrichissantes, tant sur le plan humain que technique et pratique.

Le bilan plutôt mitigé de l'AGESCA sur l'année dernière souligne qu'il est nécessaire de prendre un peu de distance et éviter de se lancer tête baissée. Le créneau formation M.A.O. est-il déjà largement saturé compte tenu des nombreuses structures proposant des stages et des cours tout au long de l'année : écoles de musiques, associations, studios et entreprises spécialisées dans la formation professionnelle.

Cependant, Didier Merlateau et moi étions d'accord pour dire qu'il y avait quelque chose de naissant et d'effervescent dans le domaine de la musique sous Linux et qu'une demande croissante de formation et d'aide à l'installation voyait le jour.

Ces rencontres m'ont aussi permis de clarifier mes souhaits en matière de pédagogie. En effet, j'envisageais mon cours de manière très organisée, en commençant par l'installation du système, puis en suivant une logique précise. Dans l'idéal, chaque stagiaire devra apporter son ordinateur de manière à effectuer une installation et une configuration personnalisées. Mais ceci peut être un peu rébarbatif pour un premier cours. Je pense que mettre à disposition des ordinateurs préconfigurés avec tous les programmes qui seront abordés pendant la formation, permettrait aux stagiaires d'avoir une première approche pratique et de voir rapidement ce qu'il est possible de faire sous Linux. Pour la suite du cours, je ne pense pas que ce soit une mauvaise idée d'adopter un

système de modules avec chacun un programme précis. Mais il faudra sans doute les adapter en fonction des attentes des élèves. Comme me le soulignais Rolf Entgelmeier : « *ils viennent souvent pour apprendre un truc précis qui leur servira dans leur travail ou leurs projets perso »*. La difficulté sera donc de les intéresser à la totalité du programme prévu, car bien souvent lorsqu'ils ont vu la fonction ou que l'on a abordé l'aspect du logiciel qu'ils voulaient connaître, « *ils décrochent »*.

Pour que le cours soit intéressant il faut le rendre interactif et l'aborder d'un point de vue pratique car cela permet d'assimiler les notions beaucoup plus rapidement. Ceci pourra être mis en place sous la forme de courtes notions théoriques suivies de petits exercices pratiques. Puis à la fin d'un module, je proposerai un exercice complet reprenant toutes ces notions. Cela permettra aussi de faire le point sur leur apprentissage et sur ce qu'ils n'ont pas compris. Je trouve important en fin de chaque module de faire un bilan de ce que l'on vient d'apprendre, de revoir rapidement les différentes notions importantes. Cela permet de "rafraîchir la mémoire", car la durée d'un module peut être longue et ainsi de mieux se préparer à aborder la suite du cours.

Afin de garder toute l'attention des stagiaires j'essayerai au maximum de leur donner des exemples ou de leur apprendre des fonctionnalités en relation directe avec leur projet professionnel ou personnel. Ceci permettra de créer chez les stagiaires une certaine dynamique et une motivation générale. Il faudra donc au préalable bien cibler leurs attentes.

Conclusion

Dans la première partie, mon but était le suivant : présenter les logiciels libres de manière générale, leur origine et les différentes licences dérivées. Même si cette présentation est indirectement liée à ma partie pratique qui propose une réflexion sur la création d'une formation en M.A.O., il m'a semblé nécessaire d'en parler pour répondre à la question suivante : pourquoi utiliser des logiciels libres ? En effet ce choix peut paraître surprenant mais leur évolution est très prometteuse.

Les logiciels libres sont relativement récents. Ils ne sont pas encore rentrés dans les mœurs et les mentalités. Bien qu'ils prennent une place de plus en plus importante, il faudra encore du temps avant que leur utilisation soit courante. Néanmoins, chaque jour, comme en témoignent les inscriptions sur le site Internet linuxmao.org, des utilisateurs sont conquis par ce concept de partage et de liberté.

Il est vrai que le « créneau » stage M.A.O. est déjà bien occupé. Mais les logiciels libres ne sont pas encore très utilisés et les formations quasiment inexistantes. Plus il y aura d'utilisateurs, plus la demande sera importante. Ce mémoire me permet de réfléchir à un service que je pourrai apporter à tous ces nouveaux utilisateurs et me positionner sur un marché presque vierge. Mon projet visera dans un premier temps, les amateurs et les particuliers puis, lorsque le système aura atteint une certaine maturité, je m'orienterai vers les professionnels. Cet avenir sera sans doute plus proche qu'on ne l'imagine.

J'ai toujours été attiré par l'enseignement et la transmission du savoir théorique et pratique. Aujourd'hui, grâce à ce mémoire et aux rencontres que j'ai faites, je suis sur le point de monter un projet de formation et je travaille déjà en collaboration avec l'association « Le Train En Marche » pour mettre en place de petits stages et avec Didier Merlateau pour la création d'une association.

Annexes

Annexe 1 : Le Manifeste GNU

Le Manifeste GNU

Le manifeste GNU, reproduit ci-dessous, a été
écrit par Richard
Stallman au commencement du projet GNU pour encourager la
participation et le soutien de tous. Au cours des premières
années, il y a eu quelques petites mises à jour pour
tenir compte des développements, mais il nous semble maintenant
plus judicieux de le laisser en l'état.

Nous avons appris depuis qu'il y avait quelques
incompréhensions; celles-ci peuvent être
corrigées en changeant quelques mots. Des notes de bas de page
ajoutées en 1993 aident à clarifier ces points.

Pour les dernières informations sur les logiciels GNU
actuellement disponibles, veuillez vous référer à
notre site web, en particulier notre
liste de logiciels.
Pour savoir comment contribuer, consultez
http://www.gnu.org/help.

Qu'est ce que GNU? Gnu N'est pas Unix!

GNU, l'acronyme de GNU's Not Unix (GNU n'est pas Unix), est le nom du système complet de logiciels compatible Unix que j'écris pour pouvoir le donner librement à tous ceux qui en auraient besoin. (1) De nombreux bénévoles m'aident. Les contributions en temps, en argent, en logiciels et en matériel sont les bienvenues.

Pour l'instant, nous avons un éditeur de texte, Emacs, utilisant le Lisp pour écrire des commandes d'édition, un débogueur, un générateur d'analyseurs syntaxiques compatible avec YACC, un éditeur de liens, et environ trente-cinq autres utilitaires. Un shell (un interprète de commandes) est presque terminé. Un nouveau compilateur C portable et optimisé s'est compilé lui-même et devrait être disponible cette année. Un premier noyau existe, mais nécessite plus de fonctions pour émuler Unix. Quand le noyau et le compilateur seront terminés, il sera possible de distribuer un système GNU propice au développement. Nous utiliserons TeX comme formateur de textes, mais un nroff est en cours de développement. Nous utiliserons aussi le système libre et portable X Window System. Par la suite, nous ajouterons un Common Lisp portable, le jeu Empire, un tableur et des centaines d'autres choses, plus une documentation en ligne. Nous espérons fournir, finalement, tout ce qui peut être utile et qui est normalement inclus dans un système Unix et plus encore.

GNU pourra exécuter des programmes Unix mais ne sera pas identique à Unix. Nous ferons toutes les améliorations que nous jugerons appropriées, en nous fondant sur nos

expériences avec d'autres systèmes d'exploitation. En particulier, nous prévoyons d'avoir des fichiers avec des noms longs, des numéros de version de fichier, un système de fichiers à tolérance de panne, éventuellement un système de complétion des noms de fichiers, un dispositif d'affichage indépendant du terminal, et peut-être, finalement, un système de fenêtrage fondé sur Lisp, au travers duquel plusieurs programmes Lisp et autres programmes Unix pourront partager un écran. Le C et le Lisp seront tous les deux disponibles comme langages de programmation système. Nous essayerons de supporter UUCP, MIT Chaosnet, et les protocoles de l'Internet pour la communication.

Initialement, GNU vise les machines de classe 68000/16000 avec de la mémoire virtuelle, car ce sont les machines les plus simples pour le faire fonctionner. Nous laissons l'effort supplémentaire pour l'adapter sur de plus faibles machines à ceux qui voudront l'utiliser sur celles-ci.

Pour éviter d'horribles confusions, veuillez bien prononcer le 'G' de 'GNU' quand vous parlez de ce projet.

Pourquoi j'ai dû écrire GNU

J'estime que la Règle d'or est que, si j'aime un programme, je dois le partager avec d'autres qui aiment ce programme. Les éditeurs de logiciels cherchent à diviser et à conquérir les utilisateurs, en interdisant à chacun de partager avec les autres. Je refuse de rompre la solidarité avec les autres utilisateurs de cette manière. Je ne peux pas, en mon âme et conscience, signer un accord de non-divulgation ou une licence de logiciels. Pendant des années, j'ai oeuvré au sein du Laboratoire d'Intelligence Artificielle du MIT pour résister à ces tendances, mais finalement, ils sont allés trop loin : je ne pouvais pas rester dans une institution où de telles choses avaient lieu contre ma volonté.

Pour pouvoir continuer à utiliser les ordinateurs en accord avec ma conscience, j'ai décidé de rassembler un ensemble suffisant de logiciels libres, pour pouvoir me débrouiller sans logiciels non libres. J'ai démissionné du laboratoire d'Intelligence Artificielle pour que le MIT ne puisse invoquer toutes les excuses légales pour m'empêcher de distribuer GNU librement.

Pourquoi GNU sera compatible avec Unix

Unix n'est pas pour moi le système parfait, mais il n'est pas trop mauvais. Les fonctions essentielles d'Unix semblent être les bonnes, et je pense pouvoir compléter ce qui manque à Unix sans les gâcher. Et un système compatible Unix serait commode à adopter par de nombreuses personnes.

Disponibilité de GNU

GNU n'est pas dans le domaine public. Tout le monde aura le droit de modifier et redistribuer GNU, mais aucun distributeur ne pourra restreindre ces futures redistributions. C'est-à-dire que des modifications propriétaires seront interdites. Je veux être sûr que toutes les versions de GNU restent libres.

Pourquoi beaucoup de programmeurs veulent contribuer.

J'ai rencontré beaucoup de programmeurs enthousiasmés par GNU et qui souhaitaient contribuer.

De nombreux programmeurs sont insatisfaits de la commercialisation de logiciels systèmes. Il se peut que cela leur permette de gagner plus d'argent, mais cela les amène forcement à se sentir en conflit avec les autres programmeurs en général, plutôt que d'être camarades. L'acte fondamental d'une amitié entre des programmeurs est le partage des programmes; les arrangements commerciaux typiquement utilisés de nos jours interdisent aux programmeurs de considérer les autres comme des amis. L'acheteur de programmes doit choisir entre l'amitié et l'obéissance à la loi. Naturellement, un grand nombre décident que l'amitié est plus importante. Mais ceux qui respectent la loi se sentent souvent mal à l'aise face à ce seul choix. Ils sont désabusés et pensent que programmer n'est qu'une façon de gagner de l'argent.

En utilisant GNU plutôt que des programmes propriétaires, nous pouvons être amicaux envers tout le monde tout en respectant la loi. De plus, GNU est une source d'inspiration et une bannière sous laquelle d'autres peuvent nous rejoindre dans le partage. Ceci peut nous procurer un sentiment d'harmonie, impossible à atteindre avec des logiciels qui ne sont pas libres. Pour environ la moitié des programmeurs avec lesquels j'ai discuté, c'est une satisfaction importante que l'argent ne peut pas remplacer.

Comment vous pouvez contribuer

> (Actuellement, pour la liste des tâches logicielles sur lesquelles travailler, consultez la liste des tâches GNU. Pour d'autres manières de contribuer, consultez http://www.gnu.org/help/help.fr.html.)

Je demande aux fabricants d'ordinateurs de faire don de machines et d'argent. Je demande aux individus de faire don de programmes et de travail.

Une des conséquences à laquelle vous pouvez vous attendre si vous donnez des machines, c'est que GNU tournera dessus très rapidement. Les machines doivent être complètes, prêtes à l'utilisation, sans besoin de système particulier de climatisation ou d'alimentation.

J'ai trouvé de nombreux programmeurs impatients de contribuer à mi-temps pour GNU. Pour la plupart des projets, un tel travail distribué à temps partiel serait très difficile à coordonner; les diverses parties codées indépendamment ne fonctionneraient pas ensemble. Mais ce problème n'existe pas dans le cas du projet de remplacement d'Unix. Un système Unix complet contient des centaines d'utilitaires, chacun étant documenté séparément. La plupart des spécifications des interfaces sont déterminées par la compatibilité avec Unix. Si chaque collaborateur peut écrire un remplacement compatible pour un seul utilitaire Unix et l'intégrer proprement à la place de l'original sur un système Unix, il s'ensuit que ces utilitaires fonctionneront ensemble sans problème. Même en faisant quelques concessions à la loi de Murphy qui créera quelques problèmes inattendus, l'assemblage de ces composants sera une tâche réalisable. (Le noyau demandera quand même une communication plus soutenue et sera réalisé par un petit groupe.)

Si je reçois des dons financiers, je pourrais embaucher quelques personnes à temps plein

ou à mi-temps. Le salaire ne sera peut-être pas très élevé par rapport au marché, mais je cherche des personnes pour lesquelles l'esprit de communauté est aussi important que l'appât du gain. Je considère que c'est une façon de permettre à quelques personnes dévouées de consacrer toutes leurs ressources au projet GNU, en leur évitant d'avoir à gagner leur vie autrement.

Pourquoi tous les utilisateurs en bénéficieront.

Une fois GNU achevé, tout le monde pourra obtenir de bons logiciels libres comme l'air (2).

Ceci représente beaucoup plus que l'économie d'une licence Unix. Cela veut dire que l'on va éviter la duplication inutile du travail de programmation. Cet effort pourra plutôt se diriger vers l'avancement du domaine informatique.

Les sources du système complet seront disponibles pour tous. Et cela aura pour résultat qu'un utilisateur qui a besoin de changer un composant du système aura toujours la liberté d'effectuer des changements lui-même, ou d'engager une personne ou une société capable d'effectuer ces changements pour lui. Les utilisateurs ne seront plus à la merci d'une seule personne ou d'une seule société qui possède les sources du programme et qui est la seule à pouvoir effectuer des changements.

Les écoles pourront fournir un milieu beaucoup plus éducatif en encourageant chaque étudiant à étudier et à améliorer le code du système. Le laboratoire informatique d'Harvard avait comme politique de n'installer aucun programme sur le système si ses sources n'étaient pas disponibles, et ils soutenaient cette politique en refusant carrément d'installer certains programmes. Cela m'a beaucoup inspiré.

Enfin, les frais engendrés par les questions d'appartenance et de droits des logiciels ne seront plus d'actualité.

Les mesures pour faire payer les licences des programmes et de leurs copies génèrent toujours un coût important pour la société en général, à cause des mécanismes nécessaires pour calculer combien (c'est-à-dire quels programmes) chacun doit payer. Et il faudrait un État policier pour appliquer parfaitement ces mesures. Prenons une station orbitale, où l'air doit être fabriqué à un coût important: facturer chaque litre inspiré peut être justifié, mais porter un masque/compteur toute la journée et toute la nuit est intolérable même si on a de quoi payer la facture. Et les caméras de surveillance placées partout pour vérifier que vous ne retirez jamais le masque/compteur seraient inacceptables. Il vaut mieux financer la fabrication de l'air avec une taxe par personne et se débarrasser des masques.

De copier tout ou des parties d'un logiciel semble aussi naturel à un programmeur que de respirer, tout aussi productif. Cela aussi devrait être libre.

Quelques objections facilement contrées aux objectifs de GNU

«Personne ne s'en servira si c'est gratuit, car cela veut dire que l'on ne peut compter sur aucun support.»

«Il faut faire payer le logiciel pour financer le service après-vente.»

Il y a des gens qui préfèrent payer pour GNU et le service plutôt que d'obtenir GNU sans

service. Une société qui propose le service uniquement à ceux qui ont obtenu GNU gratuitement, devrait être rentable (3).

Nous devons faire la distinction entre le support en termes de réel travail de programmation et le simple support d'assistance. On ne peut pas compter sur le premier de la part d'un simple revendeur. Si votre problème n'est pas suffisamment répandu, le revendeur vous enverra balader.

Si votre société a besoin d'un support fiable, la seule façon est d'avoir toutes les sources et tous les outils nécessaires. À partir de là, vous pouvez engager n'importe quelle personne qualifiée pour régler les problèmes. Vous n'êtes pas à la merci d'une seule personne. Avec Unix, le prix des sources rend cette solution inabordable pour la plupart des sociétés. Avec GNU ce sera facile. Il serait éventuellement concevable qu'il n'y ait personne de disponible, mais les modalités de distribution ne sont pas responsables de ce problème. GNU ne propose pas de régler tous les problèmes, mais seulement quelques-uns.

Pendant ce temps, les utilisateurs qui n'y connaissent rien en informatique ont besoin d'assistance et de personnes pour les aider à faire ce qu'ils pourraient faire très bien eux-mêmes si seulement ils s'y connaissaient.

De tels services pourraient être proposés par des sociétés qui ne font que des cours d'initiation et des réparations. S'il est vrai que les utilisateurs préfèrent dépenser de l'argent pour un logiciel intégrant un service après-vente, ils seront aussi d'accord pour payer simplement le service, ayant obtenu le logiciel gratuitement. Les sociétés de service se feront concurrence sur la qualité et le prix de leurs prestations. Les utilisateurs ne seront pas limités à une société particulière. En même temps, ceux d'entre nous qui n'ont pas besoin du service pourront utiliser le logiciel sans payer le service.

«On ne peut pas être connu sans publicité et il faut payer le logiciel pour financer la publicité.»

«Ca ne sert à rien de faire de la publicité si on peut obtenir le logiciel gratuitement.»

Il existe plusieurs formes de publicité gratuite ou bon marché qui peuvent être utilisées pour informer de nombreux utilisateurs au sujet de GNU. Cependant, il est peut-être vrai que l'on peut atteindre plus d'utilisateurs avec de la publicité. Si cela est vrai, une société qui fait de la publicité sur le service payant de copie et de distribution de GNU doit être suffisamment rentable pour assurer sa propre publicité et bien davantage. Ainsi, seuls les utilisateurs qui bénéficient de la publicité la payent.

En revanche, si de nombreuses personnes obtiennent GNU par leurs relations, de telles sociétés ne seraient pas rentables, et cela démontrerait que la publicité n'était pas vraiment nécessaire pour répandre GNU. Pourquoi est-ce que les partisans du libre échange ne veulent pas laisser cette décision au marché libre (4)?

«Ma société a besoin d'un système propriétaire pour être compétitive.»

GNU va retirer les systèmes d'exploitation du domaine de la concurrence. Vous ne pourrez pas être avantagé dans ce domaine, mais votre concurrent non plus. Vous pourrez rivaliser dans d'autres domaines. Si votre domaine est la vente de systèmes d'exploitation, vous n'aimerez pas GNU, et c'est tant pis pour vous. Si votre domaine est différent, GNU peut vous éviter d'être poussé dans le domaine onéreux de la vente de systèmes d'exploitation.

J'aimerais bien voir le développement de GNU financé par des dons de fabricants et utilisateurs, réduisant ainsi les coûts pour chacun (5).

«Les programmeurs ne méritent-ils pas d'être récompensés pour leur créativité?»

Si quelque chose mérite une récompense, c'est bien la contribution sociale. La créativité peut être une contribution sociale, mais seulement tant que la société est libre de profiter des résultats. Si les programmeurs méritent d'être récompensés pour la création de logiciels innovants, de même, ils méritent d'être punis s'ils limitent l'utilisation de leurs programmes.

«Un programmeur ne devrait-il pas avoir le droit de demander une récompense pour sa créativité?»

Il n'y a rien de mal à vouloir être payé pour son travail, ou à chercher à augmenter ses revenus, tant que l'on n'utilise pas de méthodes destructives. Mais les méthodes pratiquées dans le domaine du logiciel sont fondées sur la destruction. Extraire de l'argent aux utilisateurs d'un programme en limitant son utilisation est destructeur, car ces restrictions réduisent l'utilité du programme. Ce qui à son tour réduit la richesse apportée par ce programme à l'humanité. Quand le choix de limiter est délibéré, les conséquences néfastes qui en découlent sont de la destruction délibérée.

La raison pour laquelle un bon citoyen ne doit pas utiliser de telles méthodes destructrices pour augmenter sa richesse personnelle est que si tout le monde faisait de même, il y aurait un appauvrissement général dû à la destruction mutuelle. C'est ce que l'on appelle la morale kantienne, ou la Règle d'or. Puisque je n'apprécie pas les conséquences qui adviennent si tout le monde fait de la rétention d'informations, je ne dois pas trouver acceptable qu'un individu le fasse. Plus précisément, le désir d'être récompensé pour sa création ne justifie pas que l'on prive le monde en général de toute ou partie de cette créativité.

«Les programmeurs ne vont-ils pas mourir de faim?»

Je peux répondre que personne n'est forcé d'être un programmeur. La plupart d'entre nous n'arriverait pas à se faire payer pour faire des grimaces dans la rue. Mais nous ne sommes pas pour autant condamnés à passer notre vie dans la rue à faire des grimaces et à mourir de faim. On fait autre chose.

Mais c'est une mauvaise réponse, car elle accepte l'a priori de la question : c'est-à-dire que sans possession du logiciel, les programmeurs ne pourraient pas recevoir le moindre sou. C'est, soi-disant, tout ou rien.

La vraie raison pour laquelle les programmeurs ne vont pas mourir de faim est qu'il sera quand même possible pour eux d'être payés pour programmer; seulement, peut-être pas autant qu'actuellement.

La restriction des copies n'est pas la seule base des affaires du domaine des logiciels. C'est la base la plus commune, car c'est la plus rentable. Si ces restrictions étaient interdites ou rejetées par le client, les éditeurs passeraient à d'autres formes d'organisation, qui sont actuellement moins utilisées. Il y a de nombreuses façons d'organiser une entreprise.

Il est probable qu'avec ce nouveau système, la programmation serait moins rentable qu'elle ne l'est actuellement. Mais ce n'est pas un argument valable contre le changement.

Il n'est pas considéré comme injuste que les caissières gagnent ce qu'elles gagnent. Si les programmeurs gagnaient la même chose, ce ne serait pas non plus une injustice. (En pratique, ils gagneraient quand même beaucoup plus.)

«Les gens n'ont-ils pas le droit de gérer l'utilisation de leur créativité? »

«Contrôler l'utilisation que l'on fait de ses idées» revient à contrôler la vie des autres; et c'est souvent utilisé pour leur rendre la vie plus difficile.

Ceux qui ont étudié le problème de la propriété intellectuelle (6) à fond (les avocats, les juristes, etc.) soutiennent qu'il n'existe aucun droit intrinsèque à la propriété intellectuelle. Les différents droits de soi-disant propriété intellectuelle reconnus par le gouvernement ont été créés par des législations précises dans des buts bien précis.

Par exemple, le système de brevets a été établi pour encourager les inventeurs à divulguer les détails de leurs inventions. Sa raison d'être était d'aider la société plutôt que d'avantager les inventeurs. À l'époque, la durée de vie de 17 ans pour un brevet était courte par rapport à la cadence des évolutions technologiques. Puisque les brevets ne concernent que les fabricants, pour lesquels le coût et l'effort d'établir une licence sont minimes comparés à la mise en production, les brevets ne font souvent pas trop de tort. Ils ne gênent pas la plupart des individus qui utilisent des produits brevetés.

Le concept de droit d'auteur n'existait pas dans l'Antiquité, les auteurs copiaient souvent, et beaucoup, l'œuvre des autres. Cette pratique était utile, et c'est de cette seule façon que les travaux de certains auteurs ont survécu ne serait-ce qu'en partie. Le système du droit d'auteur a été créé expressément pour encourager les auteurs. Dans le domaine pour lequel ce système a été inventé, les livres, qui pouvaient seulement être copiés en imprimerie ne causait pas beaucoup de tort, et ne gênait pas la plupart des personnes qui lisaient ces livres.

Tous les droits de propriété intellectuelle ne sont que des licences accordées par la société parce que nous pensions, à tort ou à raison, que la société en général bénéficierait de ces accords. Mais dans chaque situation précise, nous devons nous demander : bénéficierons-nous vraiment d'accorder cette licence? Quels actes autorisons-nous avec cette licence?

Le cas des logiciels aujourd'hui est très différent de celui des livres il y a un siècle. Le fait que la manière la plus répandue de copier un programme est entre voisins, le fait qu'un programme contient à la fois du code source et du code binaire bien distinct, et le fait qu'un programme est utilisé plutôt que lu comme divertissement, se réunissent pour créer une situation dans laquelle celui qui applique le droit d'auteur fait du tort à la société, matériellement et spirituellement; cette personne ne devrait pas appliquer le droit d'auteur, que la loi l'y autorise ou non.

«La compétition permet de mieux faire les choses.»

Le paradigme de la compétition est une course : en récompensant le vainqueur, nous encourageons tout le monde à courir plus vite. Quand le capitalisme fonctionne réellement de cette façon, tout marche bien; mais ses partisans ont tort s'ils pensent que cela fonctionne toujours de cette façon. Si les coureurs oublient le pourquoi de la récompense, et deviennent obsédés par la victoire, quelles que soient les méthodes employées, ils risquent de trouver d'autres stratégies telles qu'agresser les autres concurrents. Si tous les coureurs s'engageaient dans un combat, ils finiraient tous en retard.

Les logiciels propriétaires et secrets sont l'équivalent moral des coureurs qui se battent. Malheureusement, le seul arbitre que l'on ait ne semble pas s'opposer aux combats ; il se contente de les réguler («Pour dix mètres parcourus, vous avez le droit de tirer un coup de feu»). Il devrait en fait séparer les combattants, et punir les coureurs qui tentent de se battre.

«Les gens s'arrêteront-ils de programmer sans l'appât du gain?»

En fait, beaucoup de gens programmeront même sans aucun bénéfice financier. La programmation exerce une fascination irrésistible pour quelques-uns, généralement ceux qui programment le mieux. Il n'y a aucune pénurie de musiciens professionnels qui continuent à jouer, même sans l'espoir de pouvoir en faire leur gagne-pain.

Mais en fait cette question, bien qu'elle soit souvent posée, ne convient pas à la situation. Les salaires des programmeurs ne disparaîtront pas mais diminueront peut-être. La question devient donc, trouvera-t-on des programmeurs qui travailleront pour une moindre rémunération? D'après mon expérience, la réponse est oui.

Pendant plus de dix ans, plusieurs des meilleurs programmeurs du monde ont travaillé au laboratoire d'Intelligence Artificielle du MIT pour un salaire bien moins important que ce qu'ils auraient touchés ailleurs. Ils étaient récompensés de plusieurs autres manières: la notoriété, le respect des autres, par exemple. Et la créativité est une récompense en soi.

Et puis la plupart sont partis pour faire le même travail pour beaucoup plus d'argent.

Les faits démontrent que les gens programmeront pour d'autres raisons que l'accumulation de richesses; mais si on leur propose beaucoup plus d'argent, ils s'y attendront finalement et l'exigeront. Les organismes qui payent moins bien ont du mal face à ceux qui payent bien, mais ils devraient pouvoir s'en sortir si les gros payeurs sont bannis.

«Nos besoins en programmeurs sont tellement importants que s'ils interdisent le partage, nous ne pouvons que leur obéir.»

La situation n'est jamais aussi désespérée au point d'être amené à obéir à une telle interdiction.

«Les programmeurs doivent bien gagner leur pain.»

À court terme, cela est vrai. Cependant, il y a de nombreuses possibilités offertes à un programmeur pour vivre décemment sans pour autant vendre le droit d'utiliser un programme. Cette façon est la plus répandue actuellement, car c'est celle qui engendre le plus de profit pour les programmeurs et les hommes d'affaires, et non parce que c'est la seule manière de gagner son pain. Vous pouvez facilement trouver d'autres manières si vous le voulez. Voici quelques exemples.

Un fabricant arrivant avec un nouvel ordinateur payera pour le portage des systèmes d'exploitation sur le nouveau matériel.

L'offre de services d'enseignement, de conseil et de maintenance peut permettre la création d'emplois.

Les personnes avec des idées nouvelles peuvent distribuer des logiciels librement (7), en demandant des dons aux utilisateurs satisfaits ou en offrant un service de conseil. J'ai déjà rencontré des personnes travaillant ainsi.

Les utilisateurs ayant des besoins en commun, peuvent créer des groupes d'utilisateurs et payer des cotisations. Un tel groupe pourrait faire appel à une société de développement pour écrire les programmes spécifiques pour ses membres.

Toutes sortes de développement pourraient être financés par une taxe sur les logiciels :

supposons que chaque personne qui achète un ordinateur doive payer x pour cent du prix en tant que taxe sur les logiciels. Le gouvernement reverserait cette somme à un organisme tel que la NSF pour subvenir au développement de logiciels.

Mais si l'acheteur fait lui-même un don au développement de logiciels, il pourra être crédité pour cette taxe. Il pourrait donner au projet de son choix, car il espérera profiter des résultats à l'achèvement du projet. Il pourra donc être exempté de la taxe si le montant de sa donation recouvre celle-ci.

Le taux de la taxe pourrait être déterminé par un vote de ceux qui la payent, pondéré par le montant de l'imposition.

Les conséquences :

- La communauté des utilisateurs soutient le développement des logiciels.
- Cette communauté décide du niveau du soutien nécessaire.
- Pour les utilisateurs qui se soucient de quels projets profitent de leur participation, ils pourront les choisir eux-mêmes.

À terme, rendre les programmes libres est un pas vers le monde d'après pénurie, quand personne ne devra travailler très dur juste pour survivre. Les gens seront libres de se consacrer à des activités ludiques telles que la programmation, après avoir, bien entendu, passé les dix heures par semaine nécessaires pour des œuvres telles que la législation, la thérapie de famille, la réparation de robots et l'exploration d'astéroïdes. Il n'y aura donc plus besoin de gagner sa vie en programmant.

Nous avons déjà beaucoup réduit la quantité de travail que la société entière doit fournir pour sa productivité, mais seulement une petite part se traduit en temps de loisirs pour les travailleurs, car beaucoup d'activités non productives sont nécessaires pour accompagner l'activité productive. Les raisons principales sont la bureaucratie et la lutte isométrique contre la concurrence. Le logiciel libre va réduire grandement ces fuites du domaine du développement logiciel. Nous devons faire cela, pour que les gains de productivité se traduisent en moins d'heures de travail pour nous.

Notes

(1) Le choix des mots ici était irréfléchi. L'idée était que personne n'aurait à payer l'autorisation d'utiliser le système GNU. Mais cela n'était pas clair, et les gens ont souvent compris que les copies de GNU devaient toujours être distribuées peu chères ou gratuitement. Cela n'a jamais été l'intention; plus tard, le manifeste mentionne pour les sociétés la possibilité de fournir un service de distribution rentable. Par la suite, j'ai appris à bien faire la distinction entre «free» dans le sens de libre, et «free» dans le sens de gratuit. Le logiciel libre est un programme que les utilisateurs ont la liberté de distribuer et de modifier. Certains utilisateurs peuvent obtenir des copies gratuitement, tandis que d'autres les paieront et si cela peut rapporter de quoi financer l'amélioration de

programmes, tant mieux. L'important est que toute personne détenant une copie a le droit de l'utiliser en collaboration avec d'autres.

(2) Voilà un autre endroit où je n'ai pas fait la distinction entre les deux définitions de «libre». La phrase telle quelle n'est pas fausse, vous pouvez obtenir des copies de logiciels GNU gratuitement, par vos amis ou par l'Internet. Mais elle suggère effectivement la mauvaise idée. [N.d.t. - En français, la distinction entre «libre» et «gratuit» est évidente.]

(3) De telles sociétés existent actuellement.

(4)Depuis dix ans, la Free Software Foundation a levé la plupart de ses fonds à travers un service de distribution, bien qu'elle soit une association plutôt qu'une société. Vous pouvez commander des logiciels, des livres, ... à la FSF.

(5) Un groupe de sociétés a réuni des fonds vers 1991 pour financer la maintenance du compilateur C de GNU.

(6) Dans les années 80, je n'avais pas encore réalisé à quel point il était déroutant de parler du «problème» de la «propriété intellectuelle». Ce terme est évidemment partial; plus subtil dans le fait qu'il mélange diverses lois disparates qui traitent de problèmes très différents. De nos jours, je presse les gens à rejeter totalement le terme «propriété intellectuelle», de peur qu'il ne conduise d'autres personnes à supposer que ces lois forment une solution cohérente. La façon d'être clair est de parler de brevets, de droits d'auteur, et de marques déposées séparément. Voir des explications plus détaillées sur la manière dont ce terme sème la confusion et le parti-pris.

(7) Par la suite, nous avons appris la distinction entre «logiciel libre» et «graticiel» («freeware»). Le terme «graticiel» signifie logiciel que vous êtes libre de redistribuer, mais généralement, vous n'êtes pas libre de l'étudier ou de changer le code source; donc la plupart d'entre eux ne sont pas des logiciels libres. Voir la page Termes prêtant à confusion, que vous devriez éviter pour plus d'explications.

Annexe 2 : Témoignages d'utilisation de Linux en M.A.O.

Introduction

Comme on a pu le voir, le monde du libre, offre un panel de possibilité qui permet de travailler convenablement dans un système de production professionnel. Il reste encore du chemin avant d'avoir tous les outils nécessaires à la hauteur des logiciels propriétaires, mais certains sont déjà largement suffisants pour faire du travail de qualité. Le logiciel Ardour, à lui tout seul permet de faire de l'enregistrement multipiste en full duplex[1] ; du mixage stéréo ou multicanal avec toutes les automations[2] possibles ; du montage son grâce à sa compatibilité avec le Broadcast Wave, similaire au format OMF ou aux EDL. Ceci permet d'importer des sessions venant d'un logiciel de montage vidéo ou d'un autre logiciel de son ; du mastering couplé avec le logiciel Jamin ; enfin, en le couplant avec le logiciel Xjadeo, il permet de travailler le son à l'image. En effet, ce dernier permet de lire une vidéo et de se synchroniser au serveur JACK. Nous avons vu aussi qu'il existe des solutions pour travailler la composition, l'édition de partition, la synthèse et le design sonore.

Linux et le monde du libre, tendent donc à devenir une réelle alternative aux modes de productions traditionnels coûteux. J'ai notamment rencontré des musiciens et autres acteurs du monde du libre, qui travaillent exclusivement avec des logiciels libres. Chaque personne, à sa propre façon de travailler la musique, en fonction de son style, du matériel qu'il possède. Le choix des logiciels et la manière dont ils vont se connecter doivent être réfléchis, mais peut aussi différer d'une composition à une autre. Voici le témoignage de deux musiciens qui expliquent leurs manières d'appréhender la musique sous Linux.

a) Xrousse (Yohann Béréziat)

C'est à Lyon que j'ai rencontré Yohann Béréziat et c'est après quelques années d'expériences qu'il a pu définir une méthode de travail dans son studio entièrement équipé de logiciels libres. Le choix des logiciels à utiliser est primordial. Comme il l'explique « j'ai tout d'abord perdu beaucoup de temps à « faire et défaire », sans savoir vraiment par où commencer ». Il faut donc dans un premier temps bien définir ce que l'on veut faire. Car il n'existe pas un logiciel qui fait tout mais une multitude qui, une fois inter-connectés répondent précisément aux besoins.

Il utilise donc une distribution Linux spécialisée dans la M.A.O. (64Studio) principalement avec Ardour, hydrogen, Jamin, Rezound et bien sûr le serveur JACK. Chanteur compositeur de musique pop-rock, son principal instrument reste la guitare et sa voix : la première étape, c'est avec un papier et un crayon pour la phase écriture parole/musique. En effet, en allumant son ordinateur, les tentations de divaguer sont trop importantes. Dans cette phase d'écriture, il essai de mémoriser immédiatement ses

1 En full-duplex l'audio est transporté simultanément en entrée et en sortie. On peut donc jouer du son et enregistrer en même temps.

2 **L'automation** est la capacité d'un logiciel à enregistrer les mouvements de potentiomètres pour les reproduire lors des écoutes ultérieures. Cela permet d'automatiser le volume, le panoramique, les paramètres d'effets, et de plugins. L'automation est très utilisée en mixage.

trouvailles intéressantes : phrases musicales, enchaînements d'accords. Il utilise pour cela un petit enregistreur portable type dictaphone. Une fois la chanson finie, peaufinée et maîtrisée, il passe à l'enregistrement de la partie rythmique grâce au logiciel Hydrogen.

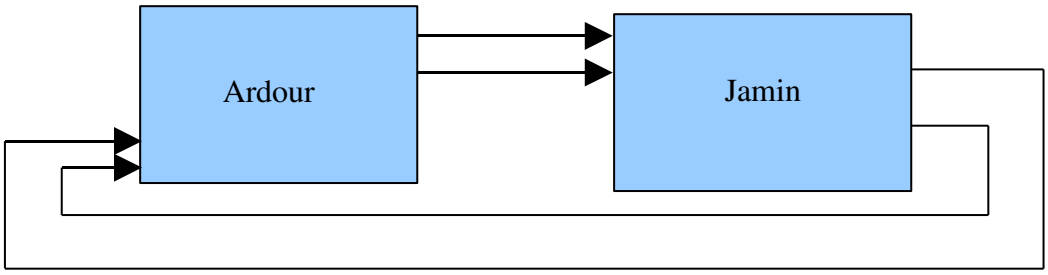
Dans cette étape, il créé plusieurs enchaînements de patterns[1] afin de poser les bases de la rythmique et donner le tempo. Elle sera retouchée et affinée une fois tous les instruments enregistrés et cela en fonction de la dynamique du morceau.

Ensuite, il connecte Hydrogen aux entrées d'Ardour et enregistre la piste batterie. Puis, il joue chaque instrument : guitare, piano, qui vont dans un premier temps poser l'harmonie et la rythmique. Il ajoute enfin les différentes pistes supplémentaires : voix, basse, synthés. Au besoin, il retouche la séquence de batterie (breaks, variations, volume et dynamique) dans Hydrogen pour coller aux instruments.

Pour le mixage, il utilise encore Ardour : ajustement des niveaux entre les pistes ; ajout d'effets reverbe sur les voix, égalisation et éventuellement des compresseurs. Une fois qu'il est satisfait, il réenregistre la piste rythmique de Hydrogen.

La phase de mastering, s'effectue avec Jamin. Ce dernier est connecté en sortie d'Ardour, puis est renvoyé dans une entrée stéréo. Comme nous l'avons déjà expliqué, Jamin se compose, d'un limiteur, d'un égaliseur 30 bandes, d'un compresseur 3 bandes, d'un visualisateur de spectre fréquentiel. Tout est réunis pour traiter la dynamique et affiner les réglages fréquentiels avant l'export final. Il n'est pas directement intégré à Ardour, c'est pour cela que l'on est obligé de réenregistrer le résultat dans une nouvelle piste d'Ardour :

[Ardour] Sortie Master--> entrée [Jamin] sortie --> entrée Stéréo pour export [Ardour]



Enfin, il exporte la nouvelle piste enregistrée en fichier wav et il convertit au format Ogg Vorbis[2] à l'aide du logiciel oggenc, ce qui permettra la diffusion sur internet ou par email.

1 Un **pattern** est un motif qui se répète. Il peut être rythmique voir mélodique. On peut enchaîner plusieurs pattern pour créer un morceau complet.

2 Vorbis est un algorithme de compression et de décompression (codec) audio numérique, sans brevet, ouvert et libre, plus performant en terme de qualité et taux de compression que le format MP3, mais moins populaire que ce dernier.

b) Hitmuri (Florent Berthaut)

Dans un registre différent, Florent Berthaut improvise sa musique en temps réel. Le choix des logiciels est donc beaucoup plus optimisé pour le jeu en live. Ancien batteur d'un groupe de métal, c'est en discutant avec d'autres qu'il a appris ce qu'était le principe du « Live looping ». Il a immédiatement désiré l'appliquer à une batterie électronique et à des effets. Partant sur des solutions matérielles, difficiles à mettre en place, il s'est rapidement tourné vers du logiciel, jusqu'à ce qu'il trouve Freewheeling sur Linux. Après deux ans de pratiques sous linux, il a sorti son troisième album « Des Leurres » en Février 2008. On peut retrouver cet album sur le site Dogmazic.net dont tous les morceaux sont sous licence Creative Commons - by-sa 2.5 ce qui signifie que l'on peut les modifier, les redistribuer et les diffuser librement à condition de citer son auteur et de garder la même licence pour les travaux dérivés.

Le matériel qu'il utilise se compose d'une batterie électronique équipée de capteurs, d'un pad électronique, d'un contrôleur midi-usb, et d'un ordinateur centre névralgique sur lequel est installée la distribution Fedora avec les paquets Planet CCRMA.

Les différents logiciels sont reliés entre eux par le serveur JACK. Les capteurs de la batterie et les pads envoient des notes MIDI vers un sampler[1] qui s'appelle Tapeutape. Il a développé ce sampler pour ses propres besoins et le distribue sous licence GPL. Les sons sont ensuite rootés vers le multi-effet Jack-Rack dont les paramètres sont modifiables via le contrôleur midi-usb. Enfin, le tout est dirigé vers un « live-looper[2] » qui permet d'enregistrer des boucles une à une et de les synchroniser sur la première. Ce logiciel s'appel Freewheeling. Dans ce logiciel, un peu rebutant au premier abord, il est possible de tout paramétrer afin d'utiliser n'importe quel contrôleur : un clavier d'ordinateur, un contrôleur midi, une pédale et même un tapis de danse. Une fois les boucles enregistrées, il peut les déclencher, les arrêter avec ses contrôleurs et en modifier les volumes.

Cette manière de travailler lui permet une grande souplesse dans la création et les morceaux sont composés en improvisation au gré de ses humeurs.

L'enregistrement est effectué avec Ardour et le mastering avec Jamin de la même manière que Yohann Béréziat.

1 Un échantillonneur, aussi connu sous l'anglicisme sampler, est un instrument de musique électronique capable d'enregistrer, d'échantillonner des sons (échantillons ou samples) et de les rejouer à la hauteur voulue.

2 Un **live-looper** permet d'enregistrer des boucles en temps réel et de les rejouer automatiquement.

Netographie

http://www.cllfst.tuxfamily.org/ : Le Club du Logiciel Libre à la Faculté des Sciences de Tunis (CLL-FST) est un groupe d'étudiants et d'enseignants, de différentes spécialités, qui a pour objectif d'encourager l'utilisation et le développement des logiciels libres parmi la communauté scientifique Tunisienne.

http://charte-zone-libre.org/ : La CHARTE ZONE LIBRE est un texte qui énumère les valeurs, principes et pratiques qu'un site, un lieu, une publication, une œuvre, un document ou toute autre Ressource consultable applique, en vue de garantir au Consultant l'accès libre, de manière permanente, à la ou les Ressources concernées, sans crainte de poursuites judiciaires, de déconnexions intempestives par un fournisseur d'accès Internet, de perquisitions ou de saisies au domicile ou ailleurs, ou encore d'accusations infondées, avec impossibilité de faire valoir ses droits, même élémentaires.

http://www.dogmazic.net/ : Dogmazic.net est le site de l'Association Musique libre. Il remplace donc *musique-libre.org*. Ce n'est pas seulement des milliers de ogg et mp3 gratuits à télécharger, c'est une invitation au partage des ressources autour des valeurs de la libre diffusion. Car cette musique gratuite est avant tout libre.

http://fr.wikipedia.org/wiki/Accueil : Le site du projet d'encyclopédie librement distribuable que chacun peut améliorer.

http://users.skynet.be/kulturo/copyleft/sommaire.html : Une introduction philosophique au copyleft.

http://www.oreilly.com/openbook/freedom/ : Livre en ligne : « Free as in Freedom » écrit par Richard Stallman

http://gnuwin.epfl.ch/articles/fr/unixpasdepanique/index.html : Livre en ligne : « Unix -- Pas de panique ! » écrit par Cyrille DUNANT et Emmanuel ECKARD.

http://zdnet/partenaires/8-fi/ : Émission « Le monde du libre, les succès de l'open source » diffusé sur Direct 8 le 19 Mars 2007

http://www.ubuntu.com/ : Site présentant la distribution Ubuntu Linux.

http://www.vnunet.fr/fr/vnunet/news/2007/03/12/open-source-linux-ubuntu : Article concernant le choix de l'Assemblée nationale pour une solution à base de logiciels libres.

http://www.aful.org/ressources/licences-libres : Article sur les différentes licences libres.

http://www.journaldunet.com/solutions/0405/040512_juridique.shtml : Cas en Allemagne d'un litige sur l'utilisation d'un logiciel libre : « *...logiciel libre n'est pas synonyme de logiciel sans droit, et que la licence libre GPL est une licence comme une autre.* **»**

http://www.destrucsaweb.com/ressources/phpmyannu/goto_33.php : Traduction de la

Licence GNU LGPL en français.

http://artlibre.org/ : Site de l'association copyleft_attitude qui présente entre autre la Licence Art Libre.

http://fr.creativecommons.org/ : Site officiel des licences Creative Commons.

http://www.loligrub.be/contrib/tlepoint/BASE/ : Linux Base d'administration pour le superutilisateur – Exposé. Adaptation de Rute User's Tutoriel and Exposition de Paul Sheer. Thierry Lepoint.

http://www.alsa-project.org/ : Site internet du projet ALSA.

http://freebob.sourceforge.net/ : Site internet du projet freebob.

http://jackaudio.org/ : Site internet de Jack Audio Connection Kit.

http://ardour.org/ : Site officiel du logiciel Ardour.

http://rezal404.org/wikini/wakka.php?wiki=IceCastDoc : Traduction français d'un tutorial pour créer son serveur Icecast.

http://www.livretdulibre.org/ed3/html/livret_libre.html : Livret sur l'utilité des logiciels libres par rapport aux logiciels propriétaires.

http://www.afdas.fr/ : Site de officiel de l'AFDAS **(**Assurance Formation Des Activités du Spectacle).

http://linuxmao.org/ : Site principal de la communauté francophone de musicien utilisant Linux.

http://www.fedora-fr.org/ : Site de la communauté francophone de Fedora.

L'enjeu des logiciels libres : Création d'une Formation M.A.O.[1] sous Linux.

Les logiciels libres sont de plus en plus utilisés. Déjà bien implantés dans certains domaines (serveurs, administration systèmes), ils commencent à se démocratiser auprès du grand public. L'Audio n'est pas épargné. En quelques années, de nombreux logiciels ont vu le jour et atteint un niveau suffisant pour la création musicale. Dans ce mémoire j'explique tout d'abord ce qu'est un logiciel libre. Puis j'expose ma réflexion sur la manière de créer et d'organiser une formation en M.A.O. sous Linux.

Mots clés:
Logiciels libres ; Open Source ; Linux ; M.A.O. ; formation ; licences libres ; Fedora ; PlanetCCRMA

The issue of free softwares : Creation of a C.A.M.[2] training for Linux.

Free softwares are more and more used. Already embedded in several domains (servers, systems administration), they are becoming more accessible among general public. Audio is not spared. In few years, many softwares grown and reach a sufficient level for musical creation. In this Master's dissertation, first I explain what free softwares are. Then I expose my reflexion on the way to create and organize a C.A.M. training for Linux.

Keywords:
Free Software ; Open Source ; Linux ; C.A.M. ; Training ; Free licences ; Fedora ; PlanetCCRMA

1 Musique Assistée par Ordinateur.
2 Computer-Aided Music or Computer-assisted Music.